NASA CONTRACTOR REPORT 159122

CARE III FINAL REPORT

PHASE I

VOLUME I

J. J. Stiffler, L. A. Bryant, L. Guccione

RAYTHEON COMPANY
SUDBURY, MASSACHUSETTS 01776

PREPARED UNDER
NASA CONTRACT NAS1-15072

FOR
NASA LANGLEY RESEARCH CENTER
HAMPTON, VIRGINIA

AIR FORCE AVIONICS LABORATORY
WRIGHT PATTERSON AIR FORCE BASE, OHIO

NOVEMBER 1979

NASA CONTRACTOR REPORT 159122

CARE III FINAL REPORT

PHASE I

VOLUME I

J. J. Stiffler, L. A. Bryant, L. Guccione

RAYTHEON COMPANY
SUDBURY, MASSACHUSETTS 01776

PREPARED UNDER
NASA CONTRACT NAS1-15072

FOR
NASA LANGLEY RESEARCH CENTER
HAMPTON, VIRGINIA

AIR FORCE AVIONICS LABORATORY
WRIGHT PATTERSON AIR FORCE BASE, OHIO

NOVEMBER 1979

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-159122		

4. Title and Subtitle	5. Report Date
CARE III Final Report, Phase 1, Volume I	November 1979
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
J. J. Stiffler, L. A. Bryant, L. Guccione	
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
Raytheon Company Sudbury, MA 01776	NAS1-15072
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Langley Research Center, Hampton, VA 23665 Wright Patterson Air Force Base Air Force Avionics Laboratory, Dayton, Ohio	14. Sponsoring Agency Code LARC, WPAFB

15. Supplementary Notes

NASA, Project Engineer, Salvatore J. Bavuso
WPAFB, Technical Monitor, Lt. Barry Baxley

16. Abstract

This report describes the work done during the first phase of a two-phase effort to develop a computer program to aid in assessing the reliability of fault-tolerant avionics systems. The overall effort consists of five major tasks: 1) Establish the basic requirements that must be satisfied if the program is to achieve its overall objective. 2) Define a general program structure consistent with these requirements. 3) Develop and program a mathematical model relating the reliability of a fault-tolerant system to the (not necessarily time-independent) failure rates and coverage factors characterizing its various elements. 4) Develop and program a mathematical model for evaluating the coverage (probability of successful recovery) associated with any given fault as a function of the type and location of the fault, the applicable fault detection and isolation mechanism, and the number and status of prior faults. 5) Develop and program a procedure whereby a user of these models can accurately and conveniently specify the configuration of the system to be evaluated and the constraints influencing its ability to recover from faults.

The first three of these tasks were completed during Phase One; the resulting requirements, program structure, and reliability model are discussed in detail in Volume I of this report, along with the tradeoffs and sample reliability assessments made in arriving at the approach finally taken. The Computer Program Requirements Document is contained in Volume II. This latter volume also includes several appendices containing computer print-outs and other ancillary material supporting the conclusions presented in Volume I.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Fault-Tolerant Avionics Systems, Reliability Modeling, Fault Coverage, Fault Models	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified		

*For sale by the National Technical Information Service, Springfield, Virginia 22161

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF SYMBOLS

Symbol	Definition
$A(t\|\underline{\ell})$	See Table 3.5
$A'(t\|\underline{\ell})$	See Table 3.5
$a_x(t)$	See Table 3.5
$a_{x_i}(t)$	See Table 3.5
$B_{x_i,y_j}(\underline{\ell},t)$	See Table 3.5
$b_{x_i,y_j}(\underline{\mu},\underline{\ell})$	See Table 3.4
$c_{\underline{j}\underline{\ell}}(t)$	Probability that the system recovers following a failure that takes the system from state $\underline{j}$ to state $\underline{\ell}$.
$\bar{c}_{\underline{j}\underline{\ell}}(t)$	$1 - c_{\underline{j}\underline{\ell}}(t)$
$D_{y_i}(\underline{\ell},t)$	See Table 3.5
$dy_j(\underline{\mu},\ell)$	See Table 3.4
$g_1(t,x_i)$	See Table 3.5
$g_2(t,x_i)$	See Table 3.5
$K_{\underline{\ell}}(t)$	Probability density of transition into failed state with $\underline{\ell}$ faulty units.
$\underline{L}$	A vector, the components of which indicate the numbers of faults that have occurred in each category ($\underline{L} = \ldots \ell_{x_1}, \ell_{x_2}, \ldots \ell_{y_1}, \ell_{y_2} \ldots$).
ℓ	Total number of faults that have been experienced ($\ell = \sum_x \ell_x$)

Symbol	Definition
$\underline{\ell}$	A vector, the components of which indicate the number of faulty elements in each stage ($\underline{\ell} = \ldots \ell_x, \ell_y, \ldots$).
$\underline{\ell} - \epsilon_y$	$(\ldots \ell_x, \ell_y - 1, \ell_z, \ldots)$
ℓ_x	Number of stage x modules that have experienced some fault ($\ell_x = \sum_i \ell_{x_i}$)
ℓ_{x_i}	Number of stage x modules that have experienced category x_i faults.
$M_o(\underline{\mu}, \underline{\ell})$	$M_o(\mu_p, \ell_p) M_o(\mu_m, \ell_m)$ (FTMP) (p = processor, m = memory)
$M_o(\underline{\mu}_x, \ell_x)$	Conditional probability, given μ_x latent and ℓ_x total stage-x faults, that no two modules in the same triad are faulty (FTMP).
m_x	See Table 3.4
$N_i(\underline{\mu}_b, \ell_b)$	Conditional probability, given μ_b latent and ℓ_b total bus faults, that i active buses are faulty (FTMP).
n_x	See Table 3.4
$P(\underline{\mu} \mid \underline{\ell}, t)$	See Table 3.5
$P(\underline{\mu}_x \mid \ell_x, t)$	See Table 3.5
$p(t \mid \tau, x_i, y_j)$	See Table 3.4
$p_{\underline{\ell}}(t)$	Probability that a system has sustained exactly $\underline{\ell}$ failures and is still operational at time t.
$p_{\underline{\ell}}^*(t)$	Probability that a system has sustained exactly $\underline{\ell}$ failures by time t.

Symbol	Definition
p_{tr}	Probability of recovering from a transient fault with no module loss (SIFT).
$p_1(t\|\tau,x_i)$	See Table 3.4
$p_2(t\|\tau,x_i)$	See Table 3.4
$Q_{\underline{\ell}}(t)$	Probability that a system has sustained exactly $\underline{\ell}$ failures and is no longer operational at time t.
$q(t\|\tau,x_i,y_j)$	See Table 3.4
$q_{x_i}(t)$	See Table 3.4
R(t)	Reliability; probability that the system is operational at time t.
r	Ratio of transient to permanent fault rates (SIFT).
$r_x(t)$	See Table 3.5
$r_{x_i}(t)$	See Table 3.5
s	Ratio of permanent fault plus "leaky" transient rate to permanent fault rate SIFT).
x	Stage index.
α	Rate of transition from active to benign fault states.
β	Rate of transition from benign to active fault states.
δ_{x_i}	Rate of detection of type x_i faults (FTMP).

Symbol	Definition
$\Lambda_{\underline{\ell}}(t,\tau)$	$\int_{\tau}^{t} \lambda_{\underline{\ell}}(\eta)\,d\eta$
$\lambda_{j\underline{\ell}}(t)$	Rate of occurrence of failures that take the system from state $\underline{j}$ to state $\underline{\ell}$ given that it successfully recovers from the resulting fault.
$\lambda_{\underline{\ell}}(t)$	Transition rate out of state $\underline{\ell}$ (i.e., the state in which the system has $\underline{\ell}$ failed modules).
λ_{ob}	Rate of occurrence of permanent bus faults (SIFT).
λ_{op}	Rate of occurrence of permanent processor faults (SIFT)
$\underline{\mu}$	A vector, the components of which indicate the current number of latent faults in each category ($\underline{\mu} = \ldots \mu_{x_1}, \mu_{x_2}, \ldots, \mu_{y_1}, \mu_{y_2} \ldots$)
μ_{x_i}	Number of stage x modules that currently have category x_i latent faults.
τ_o	Fault detection delay (SIFT).

1.0 INTRODUCTION

The CARE III (Computer-Aided Reliability Estimation, version three) computer program is being developed as a general-purpose reliability estimation tool for fault-tolerant avionics systems. The first CARE Program, developed at the Jet Propulsion Laboratory in 1971, provided an aid for estimating the reliability of systems consisting of a combination of any of several standard configurations (e.g. standby-replacement configurations, triple-modular redundant configurations, etc.) Non-unity dormancy factors were allowed as well as user-supplied non-unity coverage probabilities.

CARE II was subsequently developed by Raytheon, under contract to the NASA Langley Research Center, in 1974. It, like the original CARE, was based on a combinatorial reliability model. The model in this case, however, was considerably more versatile.

A simple mathematical expression was used to evaluate the reliability of any redundant configuration over any interval during which the failure rates and coverage parameters remained unaffected by configuration changes. In addition, provision was made for convolving such expressions in order to evaluate the reliability of a "dual-mode" system; i.e., a system in which a single coverage-parameter/failure-rate configuration change was allowed during the interval of interest. A coverage model was also developed to determine the various relevant coverage coefficients as a function of the available hardware and software fault detector characteristics (detection delay, scheduling interval, etc.), and the subsequent isolation and recovery delay statistics.

CARE II suffers from two limitations that make it difficult to use as a general-purpose reliability estimation tool for avionics systems. The most serious of these limitations is its two-mode restriction. In many avionics system configurations, each new failure precipitates a mode change (i.e., a failure rate or coverage coefficient change). Consequently, many operating modes are possible. While CARE II could be modified to allow this possibility, the resulting program would be cumbersome and the computer run-time excessive.

A second limitation in CARE II is the lack of a mechanism for specifying multiple success criteria; i.e., for allowing the user to indicate that there are several operational system configurations, as is almost always the case in avionics systems. Although this latter limitation could be easily remedied within the CARE II structure, the former could not. Accordingly, it was decided to develop a more general reliability estimation computer program specifically designed to overcome these limitations. The present report summarizes the accomplishments made during the first phase of this two-phase effort.

Three tasks were emphasized during phase one: requirements assessment; definition of program structure; development of the reliability model. The remaining work needed to complete the objectives of the CARE III program will be accomplished during phase two; viz: adaptation of the CARE II coverage model to satisfy CARE III requirements; development of a user interface for system configuration and success criteria specification; integration of the various program modules into a unified program structure.

The structure postulated for the CARE III program is described in section 4. In brief, the program will consist of three independent modules. CAREIN interprets user inputs defining the system structure, the system success criteria, the various fault models and coverage parameters, and generates files to be used by COVRGE and CARE3. COVRGE then translates these specifications into the coverage parameters associated with each of the various system stages and operating modes. The third program module, CARE3, operates on files generated by both CAREIN and COVRGE to produce system reliability estimates in accordance with the user-defined success criteria.

The major effort during phase one was devoted to developing and programming the reliability model to be implemented in CARE3. The results of this effort are described in detail in section 3. The selected mathematical model is based on Kolmogorov's forward equations. In a parallel effort, a detailed examination was made into techniques for obtaining solutions to multi-state Markov models. The initial impetus for this work was to develop an alternative model for CARE3 should the Kolmogorov method run into computational difficulties. The latter method, however, proved to be highly effective for the class of structures of concern here, overcoming most of the limitations (e.g., extremely large number of states, time invarient transition rates) associated with time-homogeneous Markov models. Nevertheless, the Markov investigation was continued when it became apparent that these techniques would be useful in determining coverage parameters associated with intermittent faults. (An example of this is presented in paragraph 3.3). The results of this investigation are described in an appendix to Volume II of this report.

The coverage model to be implemented in COVRGE will be an extension of that implemented in CARE II (Ref. 1). This coverage model has been modified to produce the (generally time-varying) recovery rates, as required by CARE III, rather than the recovery probabilities used in CARE II. The model has not yet been integrated into CARE III, however, nor has it been combined with intermittent fault models. (The reliability model tests described in section 3 used simplified coverage models involving either constant recovery rates or fixed recovery delays.) Completion of the coverage model and its integration into the CARE III structure is one of the first tasks to be completed during phase 2.

The major task remaining to be accomplished during phase 2 is the development of CAREIN. The intent here is to provide the user maximum flexibility in specifying the system structure, fault models, coverage parameters, success criteria, etc., in the simplest possible format. A general approach to this task is outlined in section 4 and detailed in Volume II of this report.

2.0 CARE III REQUIREMENTS ASSESSMENT

Four fault-tolerant systems were examined in an effort to characterize the class of structures CARE III will be expected to model and to estimate the kind and range of parameters needed to describe these structures. The four systems examined were: Boeing Aircraft Corporation's ARCS (Airborne Advanced Reconfigurable Computer System, Ref. 2), SIFT (Software Implemented Fault Tolerance Computer, Ref. 3) under development at SRI, International, FTMP (Fault-Tolerant Multi-Processor, Ref. 4) under development at Charles Stark Draper Laboratory and FTSC (Fault-Tolerant Spacecraft Computer, Ref. 5) under development at Raytheon. A study was made both of the structures of these systems and of the techniques used to estimate their reliability. The results of this study are briefly summarized in paragraph 2.1. Paragraph 2.2 then lists the requirements that were imposed on the CARE III reliability and coverage models as a result of this study and due to other considerations.

2.1 SUMMARY OF FINDINGS

2.1.1 SIFT

The SIFT computer system consists of a number of identical processors (containing both memory and processing elements) interconnected by several interprocessor buses.* The processors are dynamically assigned to various groups, with each group typically comprised of three processors, but in some cases as many as five. The loosely synchronized processors in each group perform the same operations on the same data and transmit

*The bus structure was changed subsequent to this investigation; the change, however, does not modify the conclusions reached here concerning CARE III requirements.

the results of these operations to each of the other processors in their group. Each processor evaluates its own health and that of the other processors in its group by comparing these results. Faulty processors and buses are identified by analyzing discrepancies in these results; reconfiguration takes place whenever a majority of processors in a group concludes that one of its elements is defective.

In CARE II terminology (Ref. 1), SIFT is comprised of two stages:* a processor stage consisting of m processors, and a bus stage comprising n buses. The system has failed by time t if fewer than m^1 processors or fewer than n^1 buses are still functioning, or if a coverage failure has occurred prior to that time.

The reliability of SIFT was estimated in Ref. 3 by using a continuous-time Markov model with time-independent transition parameters. Coverage was taken into account be defining a deterministic latency period τ_0 between the occurrency of a failure and its detection. If a second processor or bus fails during this period, a system failure is declared. Since all processors and buses are presumably always powered, the dormancy factor is assumed to be unity.

Note that the probability of a coverage failure is a function of the number of processors (buses) functioning at the time of a processor (bus) failure. That is, the probability of a second processor or bus failure, and hence a coverage failure during the τ_0-second latency period depends upon the number of processors or buses functioning at that time. Since

*The term "stage" refers to an ensemble of identical, interchangeable units.

the possibility is (not unreasonably) ignored that two bus (processor) failures occur between the time that a processor (bus) failure occurs and the time that the failure is detected, the coverage parameters associated with a processor (bus) failure are independent of the number of buses (processors) in operation at the time of the failure. Thus, the system can be modeled as a two-stage configuration, a processor stage exhibiting m-1 modes (corresponding to the different numbers of processors that could be functioning at the time of a new failure), and an (n-1)-mode bus stage. It is important to emphasize that there is no coupling between the two stages; a mode change in the processor stage does not result in a bus-stage mode change, and vice-versa. This simplifies the reliability model since each stage can be treated independently.

Transient faults in the SIFT model are, like permanent faults, of two types: processor faults and bus faults, both having time-independent rates of occurrence. Any transient fault can have one of two outcomes: with probability p_{tr} the system recovers completely; with probability $1-p_{tr}$ the system loses the afflicted bus or processor. The following events are not allowed: a transient fault occurring during a latent permanent fault; a permanent fault occurring during a still active transient; a transient fault occurring while a previous transient is still active.

2.1.2 FTMP

The FTMP is comprised of a set of processors, a set of memories, and a set of buses over which processors and memories can communicate. The processors, memories, and buses are each grouped into "triads." A processor triad consists

of three tightly coupled processors all committed to the same task; a memory triad consists of three memory modules all containing the same data; and a bus triad consists of three buses with each bus used for transmission purposes by exactly one of the three units comprising each processor or memory triad. The system is thus partitioned, at any given time, into a number of processor triads and a number of memory triads, with all processor-memory communication taking place over a common bus triad. Each processor-bus and each memory-bus interface (bus guardian unit) contains a voter that produces as an output the majority-vote of the three inputs received over the bus triad. Faulty processors, memories, or buses are identified by diagnosing the pattern of discrepancies observed at these voters.

Four different reliability models for the FTMP are described in Ref. 4. The first involves a 146-state discrete-time Markov model with time-invarient transition parameters. The states are defined by the number of detected and undetected faults in the processor modules, the memory modules, the bus system and the bus guardian units. The Markov model was kept to 146 states by identifying all system states involving more than two undetected faults or more than three total faults with the failed state. Other approximations were also made in order to obtain tractable transition parameters. Even so, the computer time needed to obtain numerical results using this model were such that reliabilities were determined for only the first second of FTMP operation.

To extend these results, a simplified 11-state Markov model was obtained by treating modules having detected failures

as though they were again operational and by assuming any combination of three or more faults cause a system failure. Numerical reliability results were then obtained for the first 40 seconds of FTMP operation using this model.

The reliability of the FTMP for longer durations was estimated using a combinatorial model to determine the probability that at least P_0 of P processors, M_0 of M memories, and B_0 of B buses are operating at time t (assuming perfect coverage) and by extrapolating the coverage failure probabilities obtained using the 11-state Markov model.

In a later investigation, the 11-state Markov model was modified to determine the effect of transient faults on the FTMP for short (100 minute) missions. The permanent failure states in the original model were replaced by intermittent failure states in which failures healed (temporarily) at a constant rate α and recurred at a constant rate δ. (Once a failure has occurred, it remains in the intermittent mode either until it is detected or until it results in a system failure.)

In all of these models, coverage was defined in terms of the probability that a second fault of a given type occurred during the exponentially distributed latency period of the fault in question.

In CARE II terminology, then, the FTMP model consists of three stages: processor, memory and bus. There are as many operating modes as there are modules, since the recovery probability is a function of the number of previous failures in each of the three stages. Thus, the three stages are "coupled" in that the coverage associated with a fault in

stage i depends, in part, on the absence of faults in stage $j \neq i$ during the latency period.

2.1.3 ARCS

The ARCS system involves a computer stage (consisting of three or four identical computers), several sensor stages, and several servo (actuator) stages. The non-internally-redundant computers accept information from their associated sensors, interchange this information over cross-channel buses, and generate signals to their associated servo systems. The outputs of the (generally three) servos comprising a given stage are voted on by a mechanical voting mechanism assumed to have complete first-failure fault tolerance.

The computers use a combination of hardware and software techniques to monitor their own performance and that of their associate computers, and to identify defective sensors and servos. Reconfigurations (following which, for example, a servo is deactivated, or the outputs of some sensor or computer are ignored) are effected through information passed back and forth among the ARCS computers.

The ARCS system was modeled in Ref. 2 by breaking it up into stochastically independent stages and then representing each stage with a continuous-time, constant-parameter Markov model of up to ten states. The coverages used in deriving the Markov transition parameters were estimated, in some cases, by testing actual devices using a randomly selected subset of possible faults; in other cases, coverage probabilities were simply postulated since no data were available.

The ARCS reliability model took into account the peripheral devices (sensors and servos) as well as the central computer. The ARCS architecture is such that a failure in a redundant module in one of its stages may cause the function of a module in one or more of its other stages to be lost as well. Accordingly, provision was made whereby the user could specify a "dependency" relationship among the various stages of the ARCS configuration; i.e., the user could in effect specify more than one definition of an operational system configuration.

Transient and intermittent faults were both taken into account in that they were allowed to influence the Markov transition parameters. Transients affected these parameters to the extent that they were "leaky"; i.e., the permanent fault hazard rate was increased by a term reflecting the rate of occurrence of transients of duration exceeding some test interval T. Since the Markov model implemented in the ARCS reliability evaluation program allowed unidirectional transitions only, the effect of intermittent faults (causing transitions back and forth between two states) was approximated by calculating an "effective" unidirectional transition parameter from one of these states to the other.

2.1.4 FTSC

The FTSC (Ref. 5) is an internally redundant central processor being developed for the U.S. Air Force. It is partitioned into nine types of elements (central processing unit, memory module, direct memory access unit, serial bus interface unit, power module, timing module, configuration control unit, circumvention unit, and hardened timer)

interconnected by seven different bus networks (address bus, data bus, control bus, power bus, timing bus, interrupt bus, status bus). Each of these elements and buses is provided with redundant spares, in various configurations depending upon its complexity. (One element, the memory module, is itself internally redundant as well.)

The current FTSC reliability model is a simplified, one-mode, sixteen-stage version of CARE II. In some cases, non-unity dormancy factors were used to account for the lower failure rate of inactive and unpowered modules.

2.2 CARE III REQUIREMENTS

The emphasis in the previous section was on the techniques used to estimate the reliabilities of the systems in question. At a minimum, CARE III must provide a unified model for all four of those systems and hence reproduce, under the appropriate set of conditions, the results obtained using each of these models. This, of course, is a necessary but not a sufficient condition to place on CARE III. To be most useful, it must be flexible enough to overcome any limitations imposed by the above models (e.g., restrictive coverage models, limited fault models, etc.) and at the same time sufficiently general to allow other, as yet unspecified, fault-tolerant systems to be modeled without introducing artificial restrictions. The following paragraphs outline the requirements imposed on CARE III and explain the rationale for each of these requirements in terms of the above objectives.

1. Capability of modeling up to at least 40 stages.

Rationale: This is specified in the CARE III Statement of Work. Although none of the systems considered in paragraph 2.1 require as many as 40 stages, it is not difficult to conceive of systems that do. This requirement will be satisfied in CARE III by providing a means for concatenating independent

runs. If the coupling between stages is limited, it will in fact be possible to model an arbitrarily large number of stages by making repeated runs.

2. Multiple operating modes for each set of coupled stages.

Rationale: The operating mode of a system or subsystem is, so far as its reliability model is concerned, a function of its structure (number of units of various types that have to be operational for the system to function as specified) and its coverage parameters. If the system's structure or coverage coefficients change stochastically during its operating lifetime (e.g., if they depend upon the number of faults already incurred) such changes must be reflected in its reliability model. If a mode change in one stage precipitates a mode change in some other stage, the two stages are said to be coupled. (Deterministic structural or coverage parameter changes must, of course, also be reflected in the reliability model. Such changes are relatively easily accommodated, however, by introducing time-dependent coverage parameters and by concatenating reliability models representing the disjoint time intervals during which the system structure is invarient. Thus, such mode changes impose no new constraints provided only that the coverage parameters are allowed to be time-dependent.)

CARE II allowed only one mode change (two operating modes); the exhaustion of the spares available at any one stage could cause the system to change from, say, a dual-redundant to a single-string configuration, thereby changing both the system structure and the coverage coefficients associated with each stage. Two of the systems discussed in paragraph 2.1, however,

(SIFT and ARCS) exhibited mode changes after each new fault. Thus, the two-mode limitation of CARE II is not acceptable for CARE III.

3. Separate coverage model similar to that in CARE II but capable of handling latent and intermittent faults as well as permanent faults.

Rationale: The major advantage in keeping the reliability and coverage models distinct (as they were in CARE II) is that it allows the user to concentrate on each of these two areas relatively independently and hence simplifies the task of defining the system model. In addition, there are some significant practical advantages (cf. Section 4) in separating the reliability model, driven by infrequently occurring failures, from the coverage model reflecting the much more rapid detection, isolation and recovery events.

The need to handle both intermittent and latent faults in the coverage model is evident from the discussion in paragraph 2.1.

4. Multiple success criteria

Rationale: As ARCS clearly demonstrates, some redundant systems may be considered operational under any one of a number of possible conditions. It is therefore necessary for the user to be able to define each of those conditions and for CARE III to calculate the probability that at least one of them occurs.

5. n-point failure mechanisms ("category 3" faults)

Rationale: Most fault-tolerant systems exhibit "n-point-failure" mechanisms; i.e., sets of n failures ($n \geq 1$) that can disable the system even though spare hardware is available. If two BGUs fail in the enable mode in the FTMP, for example,

the system is potentially inoperative even though spare operational modules are available. CARE II modeled such failure mechanisms only for n = 1. Although the probability of such failures is generally a rapidly decreasing function of n, it cannot a priori be considered negligible for all n > 1. The concept of a single-point failure must therefore be generalized to take this into account.

6. Time-dependent hazard rates

Rationale: All of the reliability models considered in paragraph 2.1 assumed constant hazard rates. There are at least two reasons why it would be desirable to relax this restriction: (1) Recent data indicate that at least in some environments (space) the hazard rates are far from constant. (2) The hazard rates associated with modules having internal redundancy are not constant even if the individual component hazard rates are.

7. Transient faults

Rationale: Most faults are modeled either as permanent or intermittent, the latter actually being permanent faults that manifest themselves intermittently. Some faults may well be transient in nature, however; e.g., faults due to noise or those due to improperly validated software. In such cases, no hardware damage has occurred and, as soon as the cause of the fault disappears, the system can, in principle, function as before.

8. Non-unity dormancy factors

Rationale: Of the four models discussed in paragraph 2.1, only the FTSC model allowed non-unity dormancy factors. In some cases, it is reasonable to assume that dormant (e.g., unpowered or inactive) modules may have lower hazard rates

than active modules. Non-unity dormancy factors will be defined as follows: Let $P(t)$ be the probability that an active unit survives until time t and let $P^{\alpha}(t)$ be the probability that a dormant unit survives until time t. The exponent $\alpha \leq 1$ is the dormancy factor.

3.0 RELIABILITY MODEL DEVELOPMENT

Three basic mathematical approaches were considered for development of the reliability model: (1) Extension of the CARE II method. (2) Markov chain method. (3) A recursion technique based on Kolmogorov's forward differential equations.

The CARE II approach was rejected because of the large number of operational modes needed to model some of the fault-tolerant systems of interest. The coverage probabilities in both the SIFT and the FTMP systems are functions of the number of units still operating. Thus, each new failure effectively defines a new mode of operation. As demonstrated in the CARE II Final Report (Ref. 1), the complexity of the closed-form analytic expressions used in the CARE II model is a rapidly growing function of the number of possible operating modes. Even if transform techniques (e.g. Laplace transforms) are used to eliminate the multiple integrals found in these expressions, the model becomes intractable for systems involving more than four or five operating modes.

Some effort was made to generalize the basic CARE II equation (relating the probability of operating at time t with exactly ℓ known failures to the failure rates, coverage probabilities, number of active and spare elements, etc.) to include the case in which the coverage parameters were allowed to be functions of the number of previous failures in the stage in question. This would have, in principle, drastically reduced the number of required "system modes" since a mode change would no longer necessarily be needed to accommodate a change in the number of operating units in a given stage. This effort was abandoned, however, when it

became apparent that the cross-coupling between stage coverages (i.e., the dependence of the coverage in one stage on conditions in another stage) could also be a significant factor in some cases of interest.

The term "Markov chain" in the present context denotes the following modeling structure: The system state at any given instant is characterized by all those parameters needed to determine both the likelihood that it will experience some fault at time t and the probability that it will successfully recover from that fault. These various system states are then interrelated through a set of transition functions representing the rates at which the system state changes from any given state to any other state. (Thus, the transition functions $r_{ij}(t)$ and $r_{ji}(t)$ relating states S_i and S_j define the conditional probability densities of transitions at time t from S_i to S_j and from S_j to S_i, respectively; cf., Figure 3.1.)

The avionics systems to be modeled by CARE III are to be extremely reliable; only rare combinations of unlikely events can be permitted to cause the system to fail. Consequently, numerous parameters are needed to characterize each state and, in particular, its vulnerability to subsequent faults. Specifically, each state is defined not only by the number of faults in each of its coupled stages, but by the status of each of these faults as well. The status of a fault is defined by all those parameters needed to determine the system's vulnerability to subsequent faults (e.g., detected; undetected, benign, intermittent fault of a given type; undetected, active, intermittent fault of a given type; etc.) It should not be surprising that under these circumstances, the number of states needed to characterize a system can be extremely large. If a system

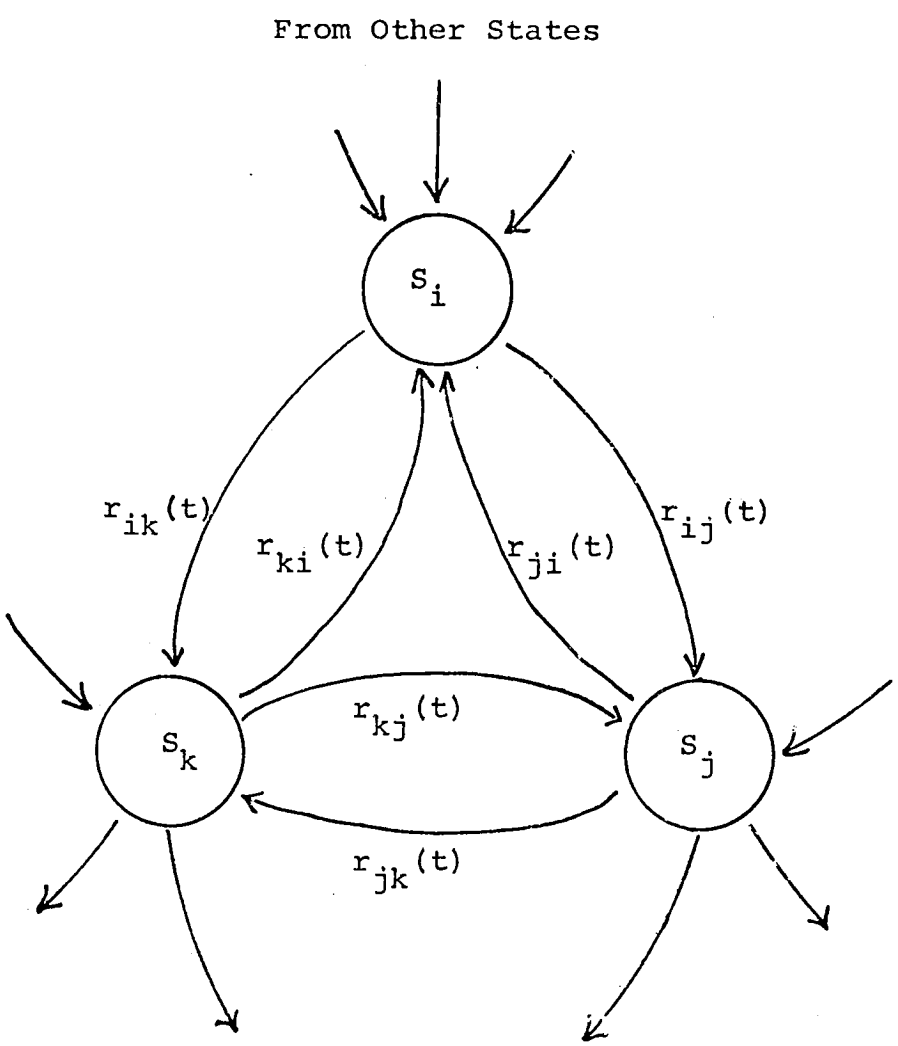


Figure 3.1

General Structure of a Markov Model

consists of n coupled stages, if the $i^{\underline{th}}$ stage can sustain as many as m faults and still be operational, and if the status of each stage-i fault can be any one of ℓ_i possibilities, the total number N of system states that have to be considered is

$$N = \prod_{i=1}^{n} \left[\sum_{j=0}^{m_i} \binom{\ell_i + j - 1}{j} \right]$$

This number can be large even for relatively small parameters ℓ_i, m_i, and n. (For example, when n = 4, and $\ell_i = 6$, $m_i = 2$ for all i, N = 614,656.)

Mathematical methods for determining the probability that a system is in any one of its Markov states at any time t are well known and particularly efficient solution techniques are available when the state transition functions $r_{ij}(t)$ are independent of t. With the Markov model just described, it is possible (although undesirably restrictive) to treat these functions as time invarient, so these mathematical methods can, in fact, be applied. Even so, these methods become computationally infeasible when the number N of states becomes large, even when advantage is taken of the fact that the number of allowed state transitions is much less than the maximum possible number, N(N-1). Since, as already noted, the number of states needed to describe systems of interest here can easily exceed 10^5, another approach was clearly needed. (Nevertheless, a thorough investigation was made into methods for efficient computer manipulation of Markov model transition matrices. This investigation was undertaken for two reasons: (1) to provide an alternative should difficulties be encountered in

developing the preferred CARE III approach; (2) to develop techniques that may be useful in implementing the CARE III coverage model. The results of this investigation are summarized in Volume II of this report.)

By far the most promising of the reliability modeling techniques examined for the class of fault-tolerant systems of concern here was one based on Kolmogorov's forward differential equations; for convenience, it will be referred to as the Kolmogorov Method. Several variations on this method were postulated and examined in detail in order to determine the most efficacious procedure for applying it to the problem at hand. The variations considered are described in the following paragraphs. Before proceeding, however, it may be useful to outline the general approach.

As already noted, the major problem with the Markov Method, as outlined, is the inordinately large number of states needed to distinguish all the various fault conditions. As also noted, these conditions can be specified in terms of two sets of parameters: 1) the number of faults in each of the coupled stages; 2) the status of each of these faults. The essence of the Kolmogorov approach is in the separate treatment of these two sets of parameters. That is, system states are used to represent only the first set of parameters; the effect of the second set of parameters is reflected implicitly in the state transition functions.

The separate treatment of the two sets of parameters needed to model fault occurrence and fault recovery has two major advantages: 1) It drastically reduces the number of states needed to represent the system (from the previously

defined number N to, in the same notation,

$$N^1 = \prod_{i=1}^{n} \left(m_i + 1\right)$$; i.e., from N = 614,656 states in the previous example to $N^1 = 81$). 2) It circumvents the serious computational difficulty presented by a model that combines in one homogeneous structure the relatively infrequent state transitions characterized by the first set of parameters (perhaps one fault/10^3 hours) and the much more frequent transitions due to fault status changes (e.g., detection rates of the order of seconds, intermittent fault transition rates of the order of minutes or less, error generation rates of the order of milliseconds).

The major disadvantages of this modeling approach are also two-fold: 1) The state transition functions are now considerably more difficult to determine. They are in effect conditioned only on time and on the number of previous failures of each type; the probability density of a transition under these conditions can be determined only by averaging over all possible values of the implicit parameters. 2) The state-transition functions are necessarily functions of time, thereby precluding from the outset the time-homogeneous Markov chain solution techniques mentioned previously.

The first of these disadvantages is reflected in a more complex coverage model than would otherwise be required. The important point here, however, is that the combinatorial and Markov techniques mentioned earlier can be applied at the coverage model level as well as at the reliability model level. Furthermore, the number of states needed to determine the

conditional transfer functions is vastly less than the number of states in an undifferentiated Markov model of the entire system. Thus, the coverage model computational effort, while greater than it would otherwise have been, is still almost negligible compared to that needed to determine the state probabilities for the system level Markov model. In effect, the model has been reduced from one having $N = n_1 \times n_2 \times \ldots \times n_\ell$ states to one having $n_1 + n_2 + \ldots + n_\ell$ states, with n_i denoting the number of relevant states given that i faults have already taken place. (The reduction is in fact more dramatic than this since much of the computational effort needed to determine the transition functions given i faults can also be used to determine these functions given $j \neq i$ faults.)

The detailed development of the CARE III coverage model will be undertaken in Phase 2 of this effort. As currently invisioned, it will combine both combinatorial and Markov techniques. The former will be used to determine the probability that a given combination of faults can, under a specific set of conditions (e.g., all faults simultaneously active) cause the system to fail; the latter will be used to determine the probability that the specified set of conditions does indeed obtain at any given time. Some specific examples of this coverage modeling approach, used during Phase one as part of the reliability model test exercise, are described in paragraph 3.3.

The second of the above-mentioned disadvantages to the modeling approach outlined here is largely overcome by basing the solution techniques on Kolmogorov's forward differential equations. The procedure for doing this is the subject of the remainder of this section.

3.1 THEORETICAL DEVELOPMENT

Let $P_{j|i}(t|\tau)$ denote the conditional probability that a system is in state j at time t given that it was in state i at time τ. Similarly, let $P_{\ell|j,\ i}(t|\eta,\tau)$ denote the conditional probability that a system is in state ℓ at time t given that it was in state j at time η and in state i at time τ. Then, clearly, for any $\tau < \eta < t$,

$$P_{\ell|i}(t|\tau) = \sum_{j} P_{j|i}(\eta|\tau) P_{\ell|j,\ i}(t|\eta,\ \tau) \qquad (1)$$

with the sum taken over all the (assumed finite number of) possible intermediate states j. (If, for all $\tau < \eta < t$, $P_{\ell|j,\ i}(t|\eta,\ \tau) = P_{\ell|j}(t|\eta)$, then equation (1) reduces to the Chapman-Kolmogorov equation for continuous-time, discrete state systems.)

It follows from equation (1) that

$$P_{\ell|i}(t + \Delta t|\tau) = P_{\ell|i}(t|\tau) P_{\ell|\ell,\ i}(t + \Delta t|t,\ \tau) + \sum_{j \neq \ell} P_{j|i}(t|\tau) P_{\ell|j,\ i}(t + \Delta t|t,\ \tau) \qquad (2)$$

Let

$$\lambda_{\ell|i}(t|\tau) = \lim_{\Delta t \to 0} \frac{1 - P_{\ell|\ell,\, i}(t + \Delta t|t,\, \tau)}{\Delta t}$$

and

$$c_{j\ell|i}(t|\tau)\lambda_{j\ell|i}(t|\tau) = \lim_{\Delta t \to 0} \frac{P_{\ell|j,\, i}(t + \Delta t|t,\, \tau)}{\Delta t}$$

(The reason for this latter notation will become apparent shortly.) Then, rearranging terms in equation (2), dividing by Δt and taking the limit as $\Delta t \to 0$ yields

$$\frac{\partial P_{\ell|i}(t|\tau)}{\partial t} = -P_{\ell|i}(t|\tau)\lambda_{\ell|i}(t|\tau) + \sum_{j \neq \ell} P_{j|i}(t|\tau)c_{j\ell|i}(t|\tau)\lambda_{j\ell|i}(t|\tau) \tag{3}$$

This set of equations is a form of the Kolmogorov forward equations. It differs from the more conventional form in that the transition parameters $c_{j\ell|i}(t|\tau)\lambda_{j\ell|i}(t|\tau)$ are also functions of the initial state i of the system at time τ. If the notation indicating the condition that the system be in state i at time τ is suppressed, equation (3) can be expressed in the more convenient form

$$\frac{dP_{\ell}(t)}{dt} = -P_{\ell}(t)\lambda_{\ell}(t) + \sum_{j\neq\ell} P_j(t)c_{j\ell}(t)\lambda_{j\ell}(t) \qquad (4)$$

It must be remembered in the ensuing discussion, however, that the transition parameters may also be functions of the initial conditions.

Four recursive reliability modeling methods based on Kolmogorov's forward equation, equation (4), were investigated in an effort to find the most suitable application of this result to the class of problems of concern here. These four methods are described in the following paragraphs.

3.1.1 DIFFERENCE EQUATION FOR RELIABILITY

Let $P_{\ell}(t)$ denote the probability that the system is operating at time t having undergone exactly ℓ failures. (If it is necessary to distinguish between different types of failures, ℓ will actually be a vector; e.g. ℓ = (i, j, k) indicating i failures of type 1, j of type 2 and k of type 3.) Let $\lambda_{\ell}(t)$ denote the rate at which failures occur given that the system has sustained ℓ failures by time t. Let $\lambda_{j\ell}(t)$ denote the rate of occurrence of failures that would, if coverage were perfect, lead from state j to state ℓ (i.e., from the state characterized by j failures to that characterized by ℓ failures).

Then

$$\sum_{\ell} \lambda_{j\ell}(t) = \lambda_j(t)$$

with the sum taken over all states ℓ which can be reached in one transition from state j. Finally, let $c_{j\ell}(t)$ denote the coverage probability associated with a failure which would, in the event of perfect coverages, cause a transition from state j to state ℓ. (The coverage associated with a failure occurring when the system is in state j is therefore

$$c_j(t) = \sum_{\ell} c_{j\ell}(t)\lambda_{j\ell}(t)/\lambda_j(t)$$

with the range of summation and the term $\lambda_j(t)$ as previously defined.)

With these definitions, equation (4), rewritten in difference-equation form

$$P_\ell(t + \Delta t) = P_\ell(t)(1 - \lambda_\ell(t)\Delta t) + \sum_j P_j(t)c_{j\ell}(t)\lambda_{j\ell}(t)\Delta t \tag{5}$$

defines a recursion, on both t and ℓ, on the probabilities $P_\ell(t)$. The probability that the system is successfully operating at time t is then just

$$R(t) = \sum_{\ell \varepsilon L} P_\ell(t) \tag{6}$$

with the summation taken over all allowable states ℓ.

Actually, equation (5) defines a recursion on ℓ only if the states can be suitably ordered. This is the case, for example, if it is impossible to go from a state having $||\ell||$ failures (with $||\ell||$ indicating the number of failed units represented by the vector ℓ) to a state having fewer than $||\ell||$; i.e., if failed units never "heal". This would appear to eliminate transient failures from the model. This is not the case, however, if the coverage coefficients make the proper distinction between "leaky" and "non-leaky" transients.

3.1.2 DIFFERENCE EQUATION FOR UNRELIABILITY

Let $P^*_\ell(t)$ be the probability that the system would be operating in state ℓ at time t were coverage perfect, let $Q_\ell(t) = P^*_\ell(t) - P_\ell(t)$ and let $\bar{c}_{j\ell}(t) = 1 - c_{j\ell}(t)$. Then equation (5) can be rewritten:

$$Q_\ell(t + \Delta t) = Q_\ell(t)[1 - \lambda_\ell(t)\Delta t] + \sum_j [Q_j(t) + P_j(t)\bar{c}_{j\ell}(t)]\lambda_{j\ell}(t)\Delta t \tag{7}$$

and the system unreliability becomes

$$1 - R(t) = \sum_{\ell \epsilon L} Q_\ell(t) + \sum_{\ell \epsilon \bar{L}} P^*_\ell(t) \tag{8}$$

with L as previously defined and $L \cup \bar{L}$ the set of all possible states.†

---

† An interesting variation on the approach suggested by this formulation is obtained by treating all states representing system failures as terminal rather than transient states. This is equivalent to redefining $Q_\ell(t)$ as the probability that the system has failed by time t and at the time of the failure it contained exactly ℓ failed units. Since unit failures occurring after the system has failed do not in this case cause a state change, equation (7) now assumes the simpler form

$$Q_\ell(t+\Delta t) = Q_\ell(t) + \sum_j P(t)\bar{c}_{j\ell}(t)\lambda_{j\ell}(t)\Delta t$$

Now, however, the two probabilities

$$Q(t) \overset{\Delta}{=} \sum_{\ell \in L} Q_\ell(t) \quad \text{and} \quad P^*(t) \overset{\Delta}{=} \sum_{\ell \in \bar{L}} P_\ell^*(t)$$

no longer represent disjoint events and equation (8) becomes an inequality rather than an equality. That is, the probability Q(t) here is a measure of the event (A) that the system has failed by time t due to a coverage failure; P(t) measures the event (B) that ℓ units have failed by time t. Thus, $1-R(t) = P(A \cup B) = P(A) + P(B) - P(A|B)P(B) \leq P(A) + P(B)$. (It can be agreed that $P(A|B) > P(A)$; that is, the conditional probability of a coverage failure given that the total number of failures exceeds some minimum must be greater than the unconditional probability of a coverage failure. Thus, $1-R(t) = P(A \cup B) \leq Q(t) + P^*(t) - Q(t)P^*(t)$.) Since clearly $1 - R(t) \geq \max(Q(t), P^*(t))$, the fact that the events A and B are not mutually exclusive is of potential concern only when Q(t) and $P^*(t)$ are both small and of the same order of magnitude. Even in this case, the unreliability would be overestimated by at most a factor of two. The reduction in computational complexity, potentially achievable by treating each failed state as a terminal state, may well justify this small reduction in accuracy; this possibility will be explored during Phase Two of this study.

This formulation offers a significant potential advantage when, as is the situation of concern here, $R(t) \approx 1$ for all t of interest. In this case,

$$\sum_{\ell \varepsilon L} P_\ell(t) \approx 1$$

and the sum of the round-off errors obtained in calculating the individual $P_\ell(t)$ terms may well be of the order of the quantity of major interest; viz: the unreliability

$$1 - \sum_{\ell \varepsilon L} P_\ell(t).$$

Under these same conditions, however, the terms $Q_\ell(t)$ must be small for all $\ell \varepsilon L$ and the terms $P^*_\ell(t)$ must be small for all $\ell \varepsilon \bar{L}$. If the round-off error associated with each of these terms can be kept small relative to the terms themselves, it follows that the cummulative round-off error will be small compared to their sum.

3.1.3 INTEGRAL EQUATION FOR RELIABILITY

Equation (4) is a linear, first-order differential equation. This equation can be easily solved to yield:

$$P_\ell(t) = e^{-\int_0^t \lambda_\ell(\tau)\,d\tau} \int_0^t \frac{\sum_{j \neq \ell} P_j(\tau)\, c_{j\ell}(\tau)\, \lambda_{j\ell}(\tau)}{e^{-\int_0^\tau \lambda_\ell(\eta)\,d\eta}}\, d\tau \qquad (9)$$

This also can be used to define a recursion on ℓ and t. If the integrals in equation (9) are replaced by their first-order approximations:

$$\int_0^{t+\Delta t} f(\tau)d\tau \approx \int_0^t f(\tau)d\tau + f(t)\Delta t$$

and if the exponentials are replaced by the first two terms in their power-series expansions:

$$e^{-f(t)} \approx 1 - f(t)$$

equation (9) is identical to equation (5). If more sophisticated approximations are used, however, it might well be possible to achieve accuracy comparable to that attainable with the equation (5) difference equations but without the need to use such small step sizes Δt. This possibility was investigated using Simpson's rule integration for the integrals in equation (9) and using an existing exponential evaluation subroutine. The results of the two approaches are compared in Section 3.2.

3.1.4 INTEGRAL EQUATION ON UNRELIABILITY

If the substitutions described in paragraph 3.1.2 are made in equation (9), the resulting expression assumes the form:

$$Q_\ell(t) = e^{-\int_0^t \lambda_\ell(\tau)d\tau} \int_0^t \frac{\sum_{j\neq\ell}\left[Q_j(\tau) + P_j(\tau)\bar{c}_{j\ell}(\tau)\right]\lambda_{j\ell}(\tau)}{e^{-\int_0^\tau \lambda_\ell(\eta)d\eta}} d\tau \qquad (10)$$

This formulation has the same potential advantage over that represented by equation (9) as the equation (7) approach has over the equation (5) approach.

3.2 EVALUATION OF THE KOLMOGOROV RECURSION METHODS

It quickly became apparent, after only a few trial program runs, that the recursions on unreliability were decidedly superior to those based on reliability for the situations of interest here. Although the reliability recursions did yield acceptable results, considerably better results could be obtained with comparable program execution time (larger step sizes) using the unreliability recursions. Consequently, the competition was quickly reduced to one between the method described in paragraph 3.1.2 and that described in paragraph 3.1.4.

The only approximations in the recursions developed in Section 3.1 are those introduced in approximating a differential equation by a difference equation or by approximating an integral by a discrete summation. The modeling task is considerably simplified, however, if one other approximation is made in these formulations. This approximation involves the determination of the coverage coefficients $c_{j\ell}(t)$.

In the examples to be considered here, the coverage coefficients are the only parameters in the reliability model recursions that are influenced by the implicit condition that the system was in state $i = 0$ at time $\tau = 0$. These terms are functions of, among other things, the probability that any of a certain subset of failures are still latent at the time of occurrence of the failure in question. Since $||\ell||$ failures took place in time t, it is clear that the

likelihood of a latent failure at time t is a generally increasing function of the ratio $||\ell||/t$. If no other conditions were imposed, it would be relatively easy to determine the probability that μ latent failures are present at time t given that the system was in state j at time t^-. There is another condition, however: the system was still operating at time t^-. This condition reduces the likelihood of certain failure sequences and hence perturbs the stochastic process characterizing failure events relative to the case when this condition does not apply. For example, the fact that the system is still operating reduces the probability that two failures occurred within a short interval of each other if a system failure would have resulted were one of these failures latent when the other took place.

It is apparent (or at least it will become apparent once specific examples are considered) that the effect of this perturbation in the stochastic failure process must be highly insignificant except, possibly, for very small values of t. in which case all failure events are extremely unlikely. Accordingly, this effect is ignored in the following formulations. The resulting distribution of latent faults is precisely that that would be found were no distinction made as to whether the system was operational or not; i.e., if no distinction was made between the state represented by the probability $P_j(t)$ and that represented by $Q_j(t)$. Since the probability of being in either of these two states is $P_j^*(t)$, therefore, the probability of a system failure at time t can be overbounded by replacing $P_j(t)$ in equation (7) or (10) by $P_j^*(t)$ and ignoring the condition on $c_{j\ell}(t)$ just discussed. Further, since ignoring this condition on the failure process

presumably results in a more favorable distribution of fault events so far as coverage at time t is concerned†, leaving $P_j(t)$ in equations (7) and (10) should result in a lower bound on the probability of system failure. In fact, as several computer runs demonstrated, the calculated system reliability is identical to six or seven decimal places regardless of whether $P_j(t)$ or $P_j^*(t)$ is used. This of course supports the contention that the ignored condition is in fact not significant.

The following paragraphs discuss the results obtained in applying the methods discussed in Section 3.1 (primarily those of paragraphs 3.1.2 and 3.1.4) to the FTMP and SIFT computers. It should be emphasized here that the purpose of these

---

†To illustrate this, consider the following simplified situation. Suppose failures can occur only at discrete instants of time (t = 0, 1, 2, ...), that no two failures can occur simultaneously, and that each failure is latent for exactly one unit of time. If a second failure occurs during the latency of a previous failure (i.e., exactly one time unit later), the system fails. Now consider $c_{2,3}(t = 8)$. If the condition that the system is still operating at time t = 7 is ignored, there are exactly $\binom{8}{2}=28$ ways in which 2 failures could have occurred in the 8 time instants t = 0, 1, ..., 7; exactly 7 of these failure sequences result in a latent failure at t = 8. The probability $\bar{c}_{2,3}(8)$ of a coverage failure is therefore 7/28 = 0.25. If the condition in question is not ignored, however, the number of possible sequences is reduced to 21, 6 of which result in a latent failure at t = 8. The probability of a coverage failure is thus increased to 6/21 = 0.286. Note that even in this extreme case, with t small (only 8 times the latency period), $||\ell||$ large (the third failure occurs after only 8 latency periods), and with all latent failures causing a system failure in the event of any other failure, the effect of the condition in question is to increase $\bar{c}$ by 14%. Under more realistic conditions, the effect on the coverage coefficients should be entirely insignificant.

exercise was not to model the computers themselves, but rather to incorporate the same general assumptions used in the previously developed models for these computers and to compare the results thus obtained with the results obtained using these earlier models.

The purpose of this effort was to judge the efficacy of the various reliability models under consideration before proceeding with their more detailed development. In order to accomplish this, it was necessary to derive analytic expressions for the coverage probabilities needed in the reliability model. This task was subsequently eliminated, so far as the user is concerned, by restructuring the reliability model. This restructuring, and the application of the restructured model to both FTMP and SIFT are described in paragraph 3.3. The following paragraphs, therefore, concentrate on the results of this reliability model comparison rather than on the derivation of expressions for $\bar{c}_{ij}(t)$.

3.2.1 APPLICATION TO FTMP - PERMANENT FAILURE CASE

The four recursions discussed in paragraphs 3.1.1, 3.1.2, 3.1.3, and 3.1.4 (henceforth to be referred to as reliability models RM1, RM2, RM3, and RM4, respectively) were first used to model the FTMP with all failures treated as permanent.

The first recursions to be programmed for this application were RM3 and RM4. For comparative purposes, an exact solution was determined analytically for the probability $P_{3,0,0}(t)$ (i.e., the probability that the system is still operating at time t after having sustained exactly three

processor failures, no memory failures and no bus failures).* This exact solution was also programmed and the result used to evaluate the accuracy of the two recursive methods. The values obtained for t = 30 seconds, for example, when the

---

*The exact solution can be expressed as follows:

$$P_{3,0,0}(t) = \left[\binom{n_p}{3}(1-e^{-\lambda_p t})^3 - 2n_p(n_p-2)A(\lambda_p,\delta_p,t) - 2n_p(n_p-3)B(\lambda_p,\delta_p,t) - 4n_pC(\lambda_p,\delta_p,t) - 2n_p(n_p-3)D(\lambda_p,\delta_p,t)\right] e^{-(n_p-3)\lambda_p t} e^{-n_m\lambda_m t} e^{-n_B\lambda_B t}$$

$$A(\lambda,\delta,t) = \frac{\lambda}{3(\delta+2\lambda)} - \frac{\lambda e^{-\lambda t}}{2(\delta+\lambda)} - \frac{\lambda^3 e^{-(\delta+2\lambda)t}}{(\delta^2-\lambda^2)(\delta+2\lambda)} + \frac{\lambda e^{-3\lambda t}}{6(\delta-\lambda)}$$

$$B(\lambda,\delta,t) = \frac{\lambda}{6(\delta+\lambda)} - \frac{\lambda^3 e^{-(\delta+\lambda)t}}{(\delta^2-\lambda^2)(\delta-2\lambda)} - \frac{\lambda e^{-2\lambda t}}{2(\delta-\lambda)} + \frac{\lambda e^{-3\lambda t}}{3(\delta-2\lambda)}$$

$$C(\lambda,\delta,t) = \frac{\lambda\delta}{6(\delta+\lambda)(\delta+2\lambda)} - \frac{\lambda^2\delta e^{-(\delta+\lambda)t}}{(\delta^2-\lambda^2)(\delta-2\lambda)} - \frac{\lambda e^{-2\lambda t}}{2(\delta-\lambda)} + \frac{\lambda^2 e^{-(\delta+2\lambda)t}}{(\delta-\lambda)(\delta+2\lambda)} + \frac{\lambda\delta e^{-3\lambda t}}{3(\delta-\lambda)(\delta-2\lambda)}$$

$$D(\lambda,\delta,t) = \frac{\lambda^2}{3(\delta+\lambda)(\delta+2\lambda)} + \frac{\lambda^2 e^{-(\delta+\lambda)t}}{(\delta+\lambda)(\delta-2\lambda)} - \frac{\lambda^2 e^{-(\delta+2\lambda)t}}{(\delta-\lambda)(\delta+2\lambda)} - \frac{\lambda^2 e^{-3\lambda t}}{3(\delta-\lambda)(\delta-2\lambda)}$$

with n_p, n_m, n_B denoting the initial number of processors, memories, and buses, λ_p, λ_m, λ_B, their respective hazard rates, and S_p the detection rate for processor faults.

initial configuration consisted of 15 processors, 9 memories and 5 buses, were:

RM3: $P_{\ell}(t) = .26330 \times 10^{-15}$

RM4: $P_{\ell}(t) = .25575 \times 10^{-15}$

Exact: $P_{\ell}(t) = .25579 \times 10^{-15}$

Similarly, with a 15 processor, 8 memory, 4 bus initial configuration, the results for t = 300 hours were:

RM3: $P_{\ell}(t) = .64394340 \times 10^{-2}$

RM4: $P_{\ell}(t) = .64384685 \times 10^{-2}$

Exact: $P_{\ell}(t) = .64384684 \times 10^{-2}$

These agreements, especially between RM4 and the exact solution are surprisingly good, particularly when it is recognized that the "exact" solution is also subject to round-off error.

The results of the comparison between RM3 and RM4 strongly favored the latter model. Since RM2 presumably has the same advantage over RM1 that RM4 has over RM3, the competition, as previously noted, was quickly narrowed to RM2 and RM4.

Table 3.1 summarizes results obtained using RM2 and RM4 with $\Delta t = t_{max}/50$, and RM2 with $\Delta t = t_{max}/100$. (A more complete listing of the results summarized here and in the following examples can be found in an appendix to this report.) As can be seen, RM2 is slightly faster than RM4 when Δt is the same in the two cases. The accuracy attainable with RM4 seems to be somewhat better than that attainable with RM2 even when the latter's step size is half (and its running time nearly double) that of the former. Note, in particular, that halving the step size in the RM2 recursion

TABLE 5.1

COMPARISON OF THREE NUMERICAL EVALUATION TECHNIQUES

Time Interval		Estimated Failure Probabilities and Running Times vs. Numerical Evaluation Technique			
Modeled	Elapsed Time From Start	Integral (50 Steps)	Difference-Eq. (50 Steps)	Difference-Eq. (100 Steps)	Vol. 2 Reference Table A2-
1000 hrs.	20 hrs.	.9115504128 E-08	.4759138922 E-08	.9013134766 E-08	1, 4, 7
1000 hrs.	1000 hrs.	.2693321948 E-01	.2693321885 E-01	.2693322063 E-01	1, 4, 7
1000 hrs.	running time	41.078 secs.	40.141 secs.	76.282 secs.	
30 secs.	30 secs.	.3410688041 E-11	.3372096536 E-11	.3391798683 E-11	2, 5, 8
30 secs.	1200 ms.	.3189783213 E-13	.1656918144 E-13	.2439316532 E-13	2, 5, 8
30 secs.	600 ms.	.8284643018 E-14	-.8526508200 E-22	.4409338212 E-14	2, 5, 8
30 secs.	running time	32.866	31.094	56.244	
800 ms.	600 ms.	.8642938477 E-14	.8421766317 E-14	.8531124275 E-14	3, 6, 9
	800 ms.	.1495029013 E-13	.1466706658 E-13	.1480876007 E-13	3, 6, 9
	16 ms.	.6686304258 E-17	.8800221654 E-33	.3349778148 E-17	3, 6, 9
	running time	32.991	30.998	56.332	

always brings the results obtained more nearly in line with those obtained using RM4. Note, too, the excellent agreement between RM4 runs having very different values of t_{max}. Specifically, the t = 600 ms. result obtained when t_{max} = 30 sec. agrees quite well with that obtained when t_{max} = 800 ms. Yet in the first instance, t = 600 ms. is the first point evaluated; in the second case, it is the 37.5th point (obtained by linear interpolation between the 37th and 38th points). This close agreement clearly is not obtained with RM2, even when Δt is halved.

As a result of these comparisons, it was concluded that RM4 is clearly the best of the reliability modeling approaches examined, and that it appears to be entirely satisfactory, in terms of accuracy, stability and computer running time, for the applications of interest.

Four computer runs were made using RM4 for purposes of comparison with results obtained by Draper in their model of the FTMP. The results of these runs, with t_{max} = 800 ms. and 30 sec. are superimposed over results obtained by Draper in Figures 3.2 and 3.3 respectively. Figure 3.4 compares Draper's results with those obtained from two RM4 runs, one with t_{max} = 10 hrs. and one with t_{max} = 1000 hrs.

The RM4 results on the whole compare well with Draper's results. The reason for the discrepancy in Figure 3.2 is not clear. It is conceivable that the discrepancy is due to a difference in the assumed conditions under which certain combinations of latent faults can cause a system failure. The fact that Draper's model treats three or more concurrent undetected failures as a system failure does not, however, appear to be sufficiently restrictive to explain the difference. In any

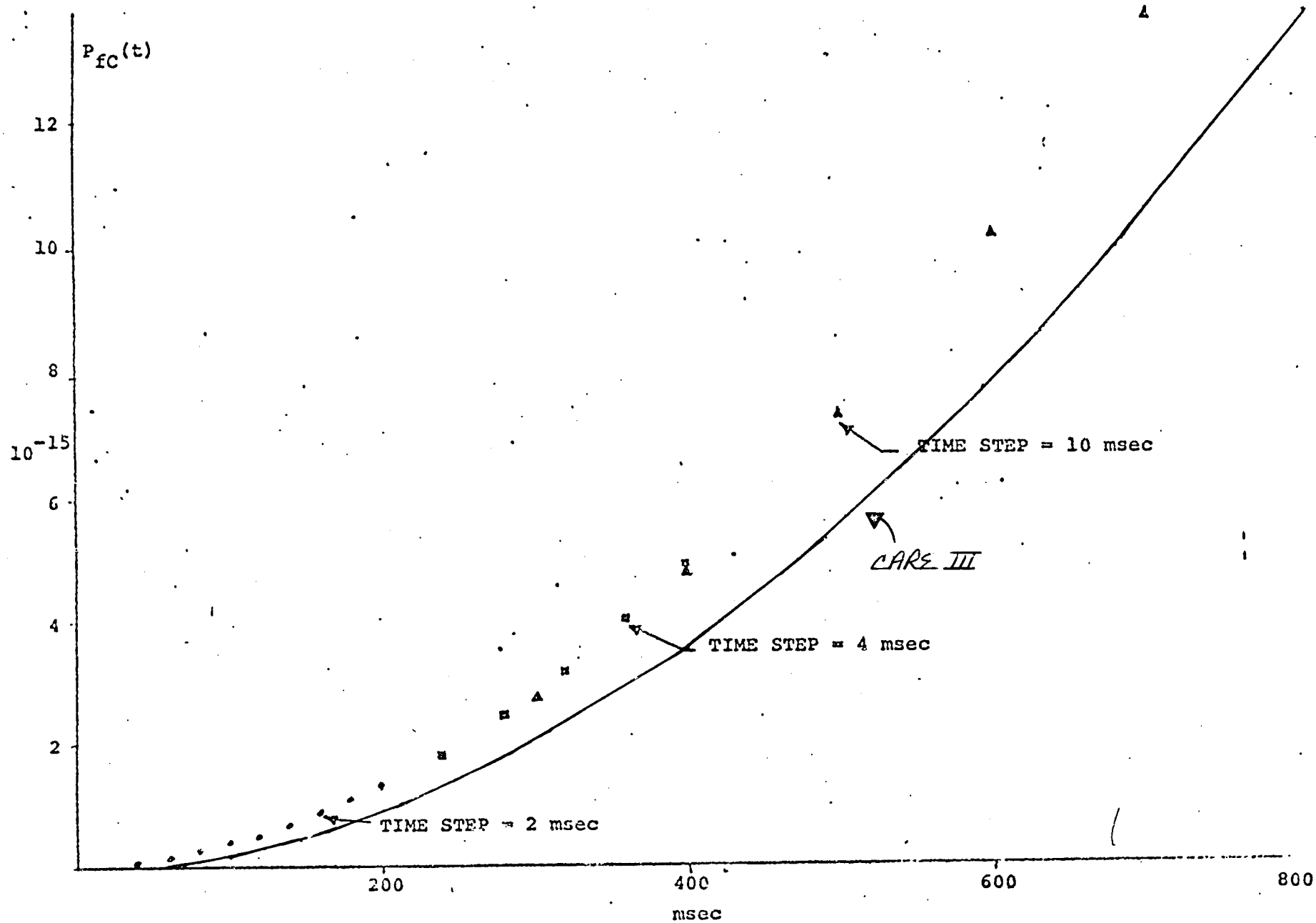


Figure 3.2 RM4 RESULTS VS. DRAPER'S 146-STATE MARKOV MODEL RESULTS

(cf. Vol. 2, Table A2-13)

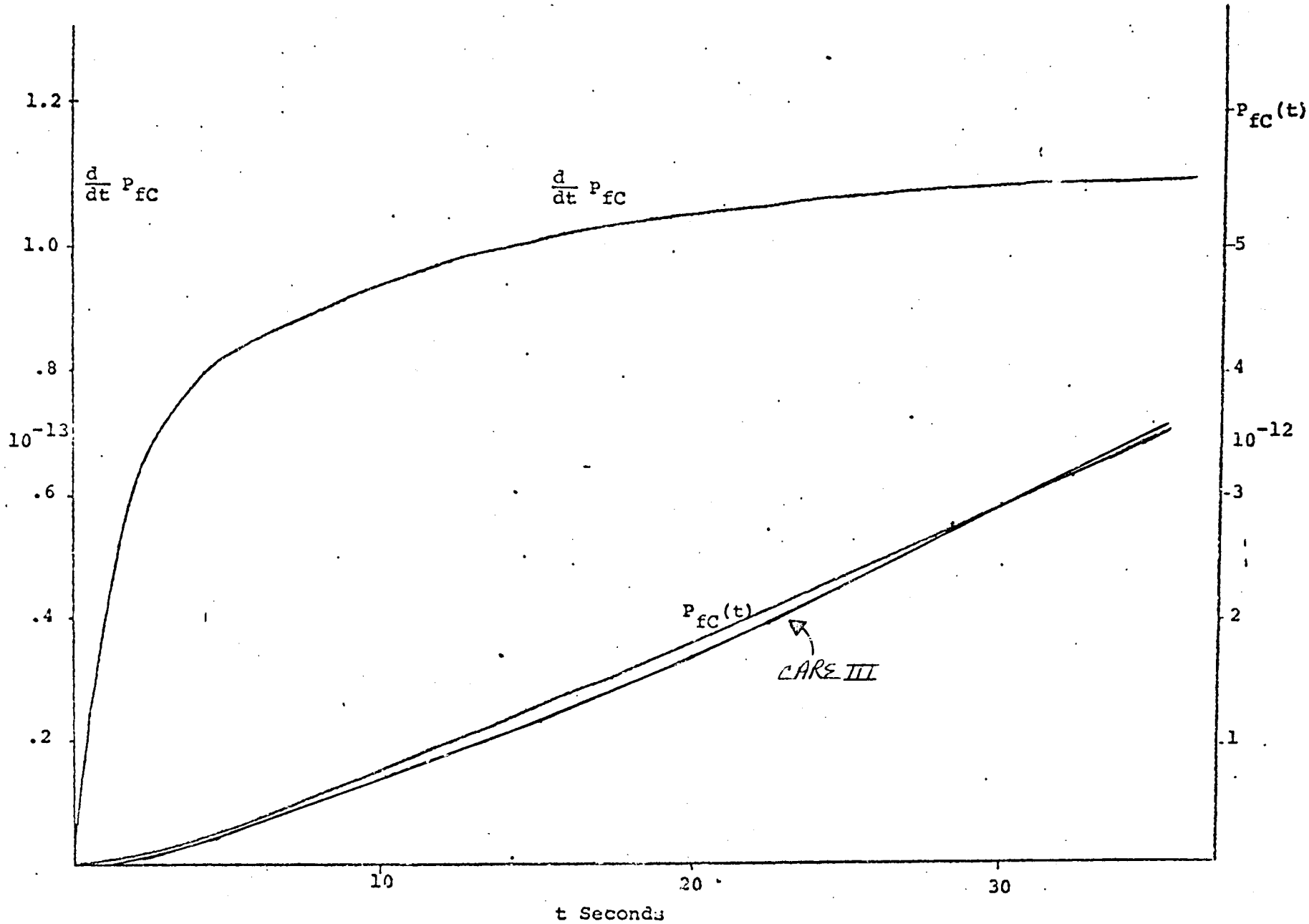


Figure 3.3 RM4 RESULTS VS. DRAPER'S 11-STATE MARKOV MODEL RESULTS
(cf. Vol. 2, Table A2-12)

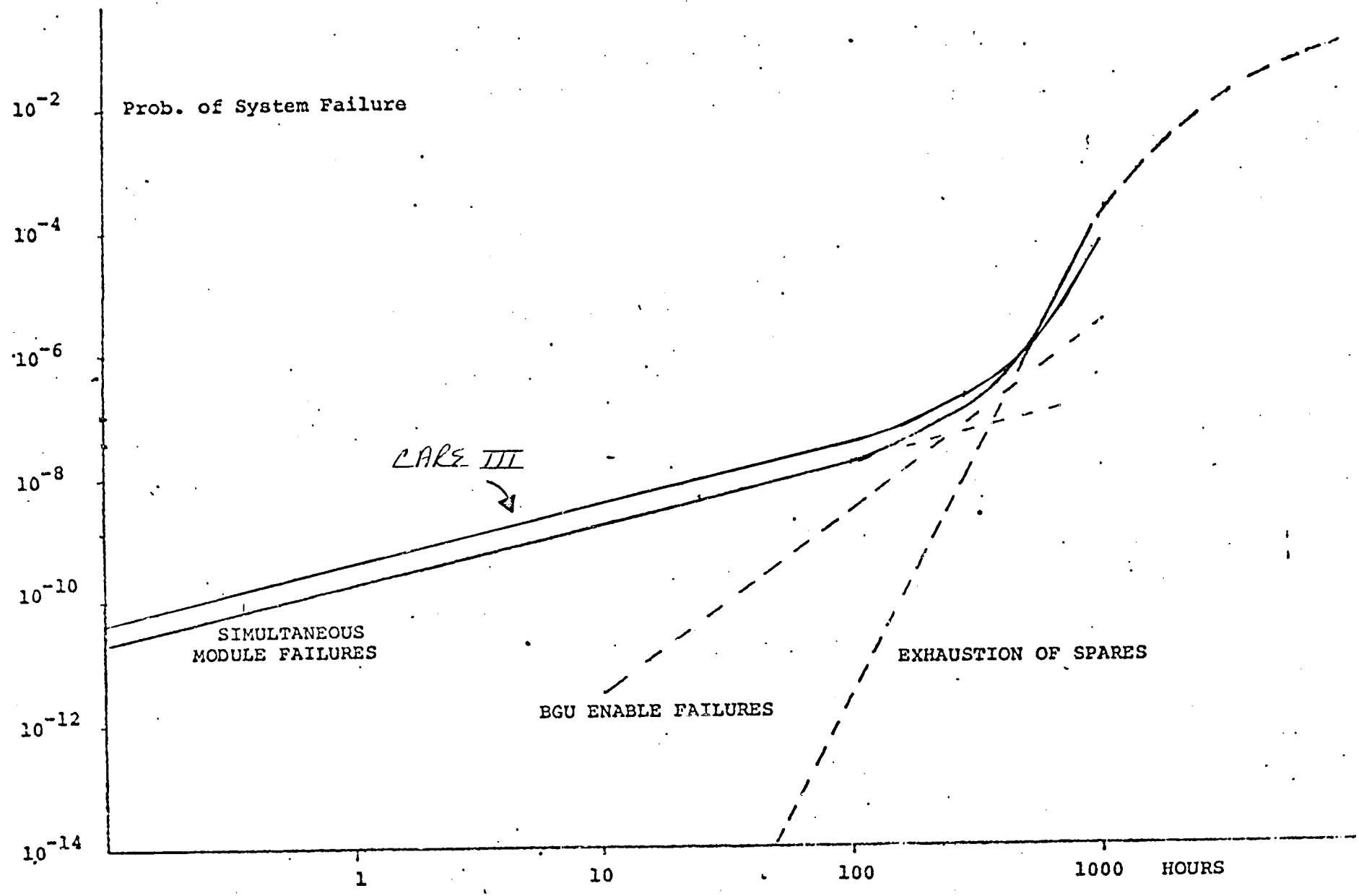


Figure 3.4 RM4 RESULTS VS. DRAPER'S EXTRAPOLATED AND COMBINATORIAL MODEL RESULTS

(cf. Vol. 2, Tables A2-10, 11)

case, the two results agree to within about 20%.

The agreement between the results obtained with RM4 and those obtained with Draper's 11-state Markov model (Figure 3.3) are remarkably good. The agreement between the two sets of results in Figure 3.4 is also quite good, the difference possibly attributable to the difficulty in plotting on a gridless graph.†

3.2.2 APPLICATION TO SIFT

Four different cases were investigated using RM4 to model SIFT. The first three cases (cases 1a, 1b and 1c) all modeled the computer in a permanent fault environment; variations were introduced in order to gauge the sensitivity of the model to what appeared to be relatively minor perturbations. Case 1a was postulated to reflect those conditions imposed in SRI's reliability model of SIFT. In that model, buses are not permitted to fail while a processor failure is still latent and processors cannot fail while a bus failure is latent. In Case 1b, this restriction is

---

†For the record, it should be mentioned that the analytical expression for coverage used for Table 3.1 was not identical to that used for Figures 3.2 3.3 and 3.4 In the former case, the recovery rate associated with a processor or memory was equated to the weighted average of the unit's recovery rate and those of its associated BGU's. In the latter cases, the slightly more cumbersome weighted average of the corresponding recovery time distributions was used. The difference in the results obtained in the two cases was small and in no way affects the conclusions gleaned from Table 3.1. The change was made before the results plotted in Figures 3.2 3.3 and 3.4 were obtained since the latter recovery model more accurately represents that postulated by Draper.

removed, but neither of these two events (bus failure during failed processor latency or vise-versa) causes a system failure. This restriction is also removed in Case 1c, but here either event does cause a system failure.

The fourth SIFT case (Case 2) involved a coverage model similar to that used in Case 1b, but the fault environment was changed to reflect SRI's transient fault model.

The results of these four investigations are summarized in Table 3.2 as are the corresponding results obtained by SRI. As can be seen, the results obtained using RM4 agree remarkably well with those obtained by SRI. The fact that the Case 1a and Case 1b results are nearly identical demonstrates that the restriction imposed by SRI in their model is indeed benign. This would be only slightly less true even if the recovery from one type of failure were adversely affected by a latent failure in a unit of the other type (Case 1c).

3.2.3 APPLICATION TO FTMP - INTERMITTENT FAULTS

The CARE III reliability model was used to estimate the reliability of the FTMP in the presence of intermittent faults. The intermittent fault model used was that defined by Draper. That is, when a fault first occurs, it is in a "bad" state, i.e., a state in which its effects are manifest. It then switches between bad states and "good" states (in which the fault is totally benign) at the constant rates β (good-to-bad) and α (bad-to-good). A fault can be detected only when it is in a bad state; the fault detection rate is then a constant δ (which may be different for the different module types).

TABLE 3.2 SIFT MODELING RESULTS
(CF. VOL. 2, TABLES A2-54 THROUGH 65)

n_p	n_b	τ SECS.	TRANS.	EXP.	CASE 1A	CASE 1B	CASE 1C	CASE 2	SRI
10	5	10	No	-8	2.486301333	2.486176900	2.762068196		2.50
9	4	10	No	-8	1.988342066	1.988242157	2.186894736		2.00
8	3	10	No	-8	4.540032421	4.540032421	4.675449311		4.56
10	5	0.1	YES	-10				2,511510104	2.55
9	4	0.1	YES	-10				2,061165614	2.10
8	3	0.1	YES	-8				3,641260227	3.65

PARAMETERS:

$\lambda_{op} = 10^{-4}/\text{hour}$

$\lambda_{ob} = 10^{-5}/\text{hour}$

$r_p = 0.1$

$r_b = 1$

$p_{tr} = 0.9$

The results obtained with the RM4 model are listed in Table 3.3 along with the results obtained by Draper using their Markov model. (To enable comparison, the parameters used in the RM4 model for α, β, δ, λ and t were precisely those used by Draper.) The column labeled CARE III, Form 1, shows the RM4 reliability predictions when no restrictions are placed on the number of faults that can be simultaneously present in the system. As can be seen, the reliabilities predicted by RM4 are generally very close to those predicted by Draper, the difference between the two predictions, however, increasing as β decreases. It was conjectured that these differences were due to two basic differences in the CARE III and Draper models: First, the Draper model did not allow more than two faults to be present at the same time, even if some of these faults were in the "good" state. Any such situation was treated as a system failure. The RM4 model places no restriction on the number of coexisting faults so long as these faults are not by themselves catastrophic (e.g., simultaneous "bad" faults in two processors in the same triad). The second difference is due to the fact that the RM4 model treats as a system failure at time t any combination of faults, first appearing at time t, that eventually cause a system failure even though the actual failure may occur at some time $t' > t$. When β is small and α large, faults spend most of their time in the good state. Thus, there can be a significant delay between the time a fault occurs and the time that it, in combination with some other intermittent fault, produces an actual failure. Since the RM4 model treated a system as being in a failed state if it contains a combination of faults that will eventually prove fatal, it is somewhat pessimistic

relative to a model in which such faults are not counted until they actually occur.

The first of these differences is thus due to a restriction on the Draper model, the second due to a restriction on the CARE III model. In order to overcome this latter restriction, a modification was made in the integrand used in the Form 1 version of RM4 described in paragraph 3.1.4. This modified version of RM4, called Form 2 and discussed in detail in paragraph 3.3, does take into account the delay between the occurrence of a fault and the resulting system failure. The results obtained with this model are also plotted in Table 3.3. As can be seen, the differences between the Form 1 and Form 2 reliability estimates can indeed be significant when $\beta << \alpha$.

Finally, in order to determine the significance of the Draper model restriction, the same restriction (more than two concurrent faults treated as a system failure) was placed on the Form 2 version of RM4. The results obtained with this restricted model (Form 2R) are tabulated in the third column of Table 3.3. A comparison of these results with those obtained by Draper (fourth column in Table 3.3) provides strong support for the conjecture concerning the differences between the Form 1 model and Draper's model.

It is believed that in most realistic situations, the difference between the reliabilities predicted by the Form 1 and Form 2 models will be insignificant. It is not possible, at this point, to conclude that this difference will be insignificant in all cases of interest, however. Accordingly, CARE III will implement both models, thereby allowing the

Table 3.3

FTMP INTERMITTENT FAULT MODEL RESULTS

(cf. Vol. 2, Tables A2-18 Through 53)

α	β	Failure Probability ($\times 10^{-8}$)			
		CARE III Form 1	CARE III Form 2	CARE III Form 2R	Draper Model
10	1	1.1181	1.1161	1.1218	1.124
10	10	1.2049	1.2041	1.2046	1.207
10	100	1.1720	1.1718	1.1720	1.174
10	1000	1.1274	1.1274	1.1275	1.129
100	1	1.0925	1.0054	1.2058	1.2073
100	10	1.9392	1.9072	1.9219	1.924
100	100	1.6614	1.6585	1.6591	1.661
100	1000	1.2182	1.2181	1.2183	1.220
000	1	0.9749	0.4239	1.4593	1.46
000	10	5.5057	3.7975	4.2295	4.22
000	100	6.2531	6.1513	6.1668	6.17
000	1000	2.1208	2.1198	2.1203	2.12

user to decide whether or not the more accurate reliability prediction afforded by Form 2 justifies its increased running time. (Form 2, when applied to FTMP, requires about three times as much CPU time as does Form 1.)

3.3 RELIABILITY MODEL STRUCTURE

Preliminary evaluation of the various reliability modeling techniques under consideration was accomplished by defining analytically the coverage functions needed for the test cases described in the previous paragraphs. This task can be arduous, however, and severely restricts the coverage model that can be accommodated. The reliability model was therefore restructured, both to increase its generality and to enable it to use coverage parameters generated by a coverage model of the sort implemented in CARE II. The new structure distinguishes among inputs defining the system structure, inputs specifying the underlying fault models and coverage-model-generated inputs characterizing the system's response to various categories of faults. This structure is described in detail in the following paragraphs.

3.3.1 SUBSYSTEM CHARACTERIZATION

The reliability model to be described here is designed to model the reliability of a subsystem consisting of some arbitrary number of stages. The system reliability is then determined by taking sums of the products of the reliabilities of appropriate sets of subsystems multiplied by the probability that no category 3 faults have occurred (cf. section 2). This last procedure, while relatively straightforward, has not yet been implemented and hence will not be discussed here.

(Combining subsystem reliabilities to determine the system reliability clearly requires knowledge of the various successful system configurations as interpreted by CAREIN. Accordingly, implementation of this operation has been deferred until after CAREIN has been more fully defined.) The discussion here concerns the task of modeling the reliability of arbitrary subsystem configurations.

Each stage in a subsystem consists of some number of identical modules or units; since the subsystem is fault-tolerant, it can presumably continue to operate successfully even after some of these units have failed. The probability that the subsystem recovers from a fault (i.e., its coverage for that fault), however, may depend upon many factors, including both the number of detected faults and the number of undetected faults in other modules in the same subsystem. (If the coverage associated with a fault in one stage is a function of the number of faults in some other stage, the two stages are said to be coupled.)

For notational convenience, each stage will be indexed by a Latin letter. Stage x, for every x, is subject to faults, each of which belongs to some category x_i, i = 1, 2, ... The subsystem state is represented by a vector $\underline{L} = (\ldots \ell_{x_1}, \ell_{x_2}, \ldots, \ell_{x_m}, \ell_{y_1}, \ell_{y_2}, \ldots)$, ℓ_{x_i} indicating the number of stage x units that have experienced a category x_i fault, etc., with each stage and each fault category thus represented. The parameter ℓ_x represents the total number of faulty stage x units, $\underline{\ell} = (\ldots \ell_x, \ell_y, \ldots)$ is a vector whose components indicate the number of faulty units of each type, and

$$\ell = \sum_x \ell_x$$

the total number of faulty units. Similarly, the vector $\underline{\mu} = (\ldots \mu_{x_1}, \mu_{x_2}, \ldots, \mu_{x_m}, \mu_{y_1}, \mu_{y_2}, \ldots)$ designates the number of latent faults in each category. (A fault is called latent if it has not yet been isolated.)

In addition to the preceding categorization, faults are also classified in accordance with their effect on the subsystem of concern at the time of their occurrence. Specifically, faults are divided into three classes: (1) Subcritical faults. A fault is said to be subcritical if it, by itself, cannot cause a subsystem failure in the absence of subsequent faults (e.g., the first processor fault in SIFT or FTMP). (2) Critical faults. A fault is called critical if it, in combination with a pre-existing latent fault, may eventually cause the system to fail even in the absence of subsequent faults (e.g., certain processor faults in SIFT or FTMP while a previous fault is still undetected). (3) Supercritical faults. A fault is designated supercritical if its occurrence causes the subsystem to fail immediately, possibly but not necessarily, as a result of pre-existing faults (e.g., faults causing single-point failures).

If a category y_j fault is critical in the presence of a pre-existing latent category x_i fault, the subsystem is said to be in an $x_i y_j$-critical state. Such a state is possible, for example, when faults (or their effects) are intermittent in nature. Faults of this sort will be said to be either active (i.e., capable of generating errors) or benign (not active). A subsystem in an $x_i y_j$-critical state will fail in the absence of other faults, if, and only if, both faults are simultaneously active. (This statement effectively defines the terms "active" and "benign.") It will be assumed that any other fault occurring while the subsystem is in a critical state will also cause it to fail. (The significance of this assumption is discussed later.)

3.3.2 SUBSYSTEM RELIABILITY MODEL

Table 3.4 defines the inputs needed for the restructured Form 1 and Form 2 reliability models. The various inputs are divided into three categories: 1) those provided by the user in defining the subsystem configuration; 2) those defined by the user in selecting fault models; and, 3) those determined by the coverage model. Table 3.5 defines both mathematically and in words the functions of these inputs evaluated by CARE3 (cf. section 3) and used to define the integrand in the RF4 version of the Kolmogorov recursion.

The RM4 recursion can be expressed in terms of these functions as follows (cf. equation 10):

$$Q_{\underline{\ell}}(t) = \int_0^t e^{-\Lambda_{\underline{\ell}}(t,\tau)} K_{\underline{\ell}}(\tau)\, d\tau \qquad (11)$$

with $\Lambda_{\underline{\ell}}(t,\tau) = \int_\tau^t \lambda_{\underline{\ell}}(\eta)\, d\eta$. The Form 1 version of $K_{\underline{\ell}}(\tau)$ can be expressed as

$$K_{\underline{\ell}}(\tau) = \sum_{y_j} [Q_{\underline{\ell}-\varepsilon_y}(\tau) + P^*_{\underline{\ell}-\varepsilon_y}(\tau)\bar{c}_{y_j}(\tau)](n_y-\ell_y+1)\lambda_{y_j}(\tau) \qquad (12)$$

with $\underline{\ell}-\varepsilon_y = (\ldots\ \ell_x,\ \ell_y-1,\ \ell_z,\ \ldots)$ and with

$$\bar{c}_{y_j}(\tau) = D_{y_j}(\underline{\ell}-\varepsilon_y,\tau) + \sum_{x_i} B_{x_i,\, y_j}(\underline{\ell}-\varepsilon_y,\tau)\, g_1(\tau,\, x_i,\, y_i) \qquad (13)$$

Equation (11) is identical to equation (10) but with a slight change in notation to emphasize the relationship between

Table 3.4

CARE3 INPUTS

Source	Function	Definition
User: configuration description	$b_{x_i,y_j}(\underline{\mu}, \underline{\ell})$	Probability that a category y_j fault would place the system in an x_i, y_j-critical state given that the total number of faults and the number of latent faults of each category, just prior to the occurrence of the category y_j fault are defined by $\underline{\ell}$ and $\underline{\mu}$, respectively.
	$d_{y_j}(\underline{\mu}, \underline{\ell})$	Probability that a category y_j fault would be supercritical given $\underline{\mu}$ and $\underline{\ell}$.
	n_x	Number of initially functioning stage-x modules.
	m_x	Minimum number of functioning stage-x modules needed for the system or subsystem to function.
User: fault model selection	$q_{x_i}(t)dt$	Probability that a category x_i fault occurs in a given stage x module in the interval (t, t+dt).
Coverage model outputs	$p_1(t\|\tau,x_i)$	Probability that a category x_i fault is active but undetected at time t given that it occurred at time τ.
	$p_2(t\|\tau,x_i)$	Probability that a category x_i fault is benign but undetected at time t given that it occurred at time τ.

Table 3.4 (Cont.)

Source	Function	Definition
Coverage model outputs	$p(t\|\tau, x_i, y_j)$	Probability that any $x_i y_j$-critical state, entered at time τ, persists until time t (i.e., neither fault has been detected nor has a subsystem failure occurred).
	$q(t\|\tau, x_i, y_j)dt$	Probability that a system failure occurs in the interval $(t, t+dt)$ as the result of an $x_i y_j$-critical state entered at time τ.

state transitions and the fault category. (Note that the summation here is over all fault categories.) Equation (13) expresses the coverage failure probability in terms of the functions defined in Table 3.5. That is, the probability of a coverage failure is just the probability that the fault in question forces the subsystem into a supercritical state plus the probability that the fault forces it into an $x_i y_j$-critical state which eventually causes it to fail.

The Form 2 expression for $K_{\underline{\ell}}(\tau)$ is

$$K_{\underline{\ell}}(\tau) = \sum_{y_j} [Q_{\underline{\ell}-\varepsilon_y}(\tau) + P^*_{\underline{\ell}-\varepsilon_y}(\tau)(\bar{c}_{y_j}(\tau) + A(\tau|\underline{\ell}-\varepsilon_y))]\lambda_{y_j}(\tau)$$

$$+ A'(\tau|\underline{\ell})\, P^*_{\underline{\ell}}(\tau) \qquad (14)$$

Here $\bar{c}_{y_j}(\tau)$ is as defined in equation (13) but with $g_1(\tau, x_i, y_j)$ replaced by $g_2(\tau, x_i, y_j)$. This reflects the fact that in the Form 2 recursion, a subsystem failure is not counted until it actually occurs. Thus, a fault forcing the subsystem into a critical state does not actually cause the system to fail at that time unless the pre-existing fault is active. The term $A'(\tau|\underline{\ell})$ accounts for subsystem failures occurring at time τ as a consequence of previously entered critical states that did not immediately cause a failure. The term $A(\tau|\underline{\ell}-\varepsilon_y)$ reflects the fact that any third fault occurring while the subsystem is in a critical state is assumed to cause it to fail.

Table 3.5

CARE3 FUNCTIONS

Function	Mathematical Expression	Definition
$r_{x_i}(t)$	$1 - \int_0^t q_{x_i}(\tau)d\tau$	Probability that a given stage x module has not experienced a category x_i fault by time t
$r_{\underline{x}}(t)$	$\prod_i r_{x_i}(t)$	Reliability of a stage x module
$\lambda_{x_i}(t)$	$q_{x_i}(t)/r_{x_i}(t)$	Rate of occurrence of category x_i faults in a given operational stage x module
$\lambda_{\underline{\ell}}(t)$	$\sum_x (n_x - \ell_x) \sum_i \lambda_{x_i}(t)$	Rate of occurrence of faults in the $\sum_x (n_x - \ell_x)$ modules that are fault-free at time t^-
$a_{x_i}(t)$	$\dfrac{\int_0^t p_s(t\|\tau, x_i) r_{\underline{x}}(\tau) \lambda_{x_i}(\tau)d\tau}{1 - r_{\underline{x}}(t)}$	Probability that a given stage x module has a category x_i latent fault at time t given that it has experienced some fault by time t
	$[p_s(t\|\tau, x_i) = p_1(t\|\tau, x_i) + p_2(t\|\tau, x_i)]$	

Table 3.5 (Cont.)

Function	Mathematical Expression	Definition
$a_x(t)$	$\sum_i a_{x_i}(t)$	Probability that a given stage x module has a latent fault at time t given that it has experienced some fault by time t
$P(\underline{\mu}_x \mid \ell_x, t)$	$\frac{\ell_x!(1-a_x(t))^{\ell_x-\mu_x}}{(\ell_x-\mu_x)!} \prod_i \frac{a_{x_i}^{\mu_{x_i}}(t)}{\mu_{x_i}!}$	Probability that a subsystem contains μ_x stage x latent faults given that it has ℓ_x faulty stage x modules
$P(\underline{\mu} \mid \underline{\ell}, t)$	$\prod_x P(\mu_x \mid \ell_x, t)$	Probability that a system having $\underline{\ell}$ faulty modules has $\underline{\mu}$ latent faults
$D_{y_i}(\underline{\ell}, t)$	$\sum_{\underline{\mu}} d_{y_i}(\underline{\mu}, \underline{\ell}) P(\underline{\mu} \mid \underline{\ell}, t)$	Probability that a system containing $\underline{\ell}$ faults would be in a supercritical state were a category y_i fault to occur at time t
$B_{x_i y_j}(\underline{\ell}, t)$	$\sum_{\underline{\mu}} b_{x_i y_j}(\underline{\mu}, \underline{\ell}) P(\underline{\mu} \mid \underline{\ell}, t)$	Probability that a system containing $\underline{\ell}$ faults would enter an $x_i y_j$-criticalstate were a category y_j fault to occur at time t

Table 3.5 (Cont.)

Function	Mathematical Expression	Definition
$g_2(t, x_i)$	$\dfrac{\int_0^t p_1(t\|\tau, x_i) r_x(\tau) \lambda_{x_i}(\tau) dt}{a_{x_i}(t)\ (1 - r_{x_u}(t))}$	Probability that a category x_i fault is active at time t given that it is latent at time t
$g_1(t, x_i)$	$1 - [1-g_2(t, x_i)][1 - \int_t^\infty q(\tau\|t, x_i, y_j) d\tau]$	Probability, given that a system enters an $x_i y_j$-critical state at time t, that this event eventually causes a system failure
$A(t\|\underline{\ell})$	$\sum_{x_i, y_j} (n_y - \ell_y + 1) \int_0^t B_{x_i y_j}(\underline{\ell} - \varepsilon_y, \tau)$ $q_{y_j}(\tau)(1-g_2(\tau, x_i)) p(t\|\tau, x_i, y_j) d\tau$	Probability that a system having $\underline{\ell}$ faults is in a critical state at time t $(\underline{\ell} - \varepsilon_y) = (\ldots \ell_x, \ell_y - 1, \ell_z \ldots)$
$A'(t\|\underline{\ell})$	$\sum_{x_i, y_j} (n_y - \ell_y + 1) \int_0^t B_{x_i y_j}(\underline{\ell} - \varepsilon_y, \tau)$ $q_{y_j}(\tau)(1-g_2(\tau, x_i)) q(t\|\tau, x_i, y_j) d\tau$	Rate at which systems having $\underline{\ell}$ faults fail at time t due to critical fault conditions

There are several assumptions implicit in these expressions which should be noted:

1. It is assumed that a faulty module can be characterized by the first fault it experiences, although the possibility of subsequent faults is not excluded. (See, for example, the expression for $a_x(t)$ in Table 3.5.) If a second fault does occur, it could have one of three effects: a) it could shorten the latency period; b) it could cause the subsystem to fail only if the first fault is still latent; c) it could cause the subsystem to fail even if the first fault has been detected.

The first of these effects can be accounted for in the coverage model, the second by adding a term to the recursion integrand $K_{\ell}(t)$ to account for that possibility, and the third can be modeled as a "category 3" failure. It is proposed, however, to ignore the first effect and to combine the second and third effects in estimating the probability of a category 3 failure. The rationale for this is as follows: The likelihood of a second failure during the latency period of a previous failure in the same module is, in most instances, entirely negligible. In any event, the approach just described overbounds the subsystem failure probability. (Ignoring the reduced latency caused by a second fault is clearly pessimistic. Treating the second effect in a separate category results in some "double counting"; i.e., a single fault is allowed to cause the subsystem to fail twice, once as a result of a second failure in the same module and again as a consequence of a failure in some other module.) The increase in the failure probability estimate as a result of such approximations is clearly

insignificant for all cases of practical interest. Thus, while more exact expressions could be relatively easily incorporated into CARE3 and COVRGE to account for such events, their minor importance does not appear to justify the added complexity.

2. Critical states are defined only for pairs of latent faults. It is possible, for example, to define an $x_i y_j z_k$-critical state in which a failure occurs only if all three faults are simultaneously active. None of the fault-tolerant systems examined thus far, however, have exhibited such failure mechanisms. Thus, while the reliability model structure described in the preceding paragraphs could readily accommodate a more general critical-state definition, the resulting added complexity does not seem to be justified.

3. Any new fault occurring while the subsystem is in a critical state causes it to fail. In many cases this is in fact not true; an arbitrary fault does not necessarily cause the subsystem to fail even when it is in a critical state. The purpose of making this assumption was, of course, to eliminate the need to account for even more complicated fault patterns involving, for example, simultaneous $x_i y_j$- and $x_i z_k$-critical states. Once again, the probability of such events is small, and the complexity needed for more precise estimation does not seem to be justified.

(It should be noted that the restriction under discussion here is considerably less severe than the restriction that three simultaneous latent faults cause a failure, as is evident from the results in paragraph 3.2).

3.3.3 SPECIALIZATION FOR FTMP AND SIFT

The input parameters used for the FTMP and SIFT test cases discussed in paragraph 3.2 are listed in Table 3.6. The FTMP model used for intermittent faults recognized only three rather than five fault categories; in this case the input parameters are as defined in Table 3.6 but with $\ell_{p_2} = \ell_{m_2} = \mu_{p_2} = \mu_{m_2} = 0.1$.

The definition of these parameters is relatively straightforward. The functions M_0, N_0 and N_1 are just the probabilities that no two modules in any FTMP processor or memory triad both contain latent faults, that no active bus contains a latent fault, and that exactly one active bus contains a latent fault, respectively. Thus, $b_{p_1p_2}(\underline{\mu}, \underline{\ell})$ for example, is the probability that no two processors or memories in any triad contain latent faults, that no active bus contains a latent fault, and that, should a category p_2 fault occur, it would affect a processor in a triad already suffering from a latent category p_1 fault. Similarly, the parameter $b_{pb}(\underline{\mu}, \underline{\ell})$ is the probability that no two processors or memories in any triad contain a latent fault, that no active bus contains a latent fault, that all memories having latent faults and all but one processor having a latent fault use the same bus, and that that bus is the one to be affected should a bus fault occur.

One class of fault situations in the FTMP requires special consideration. Suppose one of the active buses contains a latent fault, that all processors and memories containing latent faults use that bus and that at least one processor does contain a latent fault. Then a new processor fault affecting the triad already containing one latent fault

Table 3.6a

FTMP INPUT PARAMETERS *

$$L = (\ell_{p_1}, \ell_{p_2}, \ell_{m_1}, \ell_{m_2}, \ell_b)$$

$$\underline{\ell} = (\ell_p, \ell_m, \ell_b) \quad \ell_p = \ell_{p_1} + \ell_{p_2} \qquad \ell_m = \ell_{m_1} + \ell_{m_2}$$

$$\underline{\mu} = (\mu_{p_1}, \mu_{p_2}, \mu_{m_1}, \mu_{m_2}, \mu_b) \quad \mu_p = \mu_{p_1} + \mu_{p_2} \quad \mu_m = \mu_{m_1} + \mu_{m_2}$$

Let

$$M_0(\mu_x, \ell_x) = \begin{cases} 1 & \mu_x = 0, 1 \\ \dfrac{(n_x-\ell_x+\mu_x-3)(n_x-\ell_x+\mu_x-6)\cdots(n_x-\ell_x+\mu_x-3(\mu_x-1))}{(n_x-\ell_x+\mu_x-1)(n_x-\ell_x+\mu_x-2)\cdots(n_x-\ell_x+\mu_x-(\mu_x-1))} & \mu_x > 1 \end{cases}$$

$$N_0(\mu_b, \ell_b) = \frac{(n_b-\ell_b)(n_b-\ell_b-1)(n_b-\ell_b-2)}{(n_b-\ell_b+\mu_b)(n_b-\ell_b+\mu_b-1)(n_b-\ell_b+\mu_b-2)}$$

$$N_1(\mu_b, \ell_b) = \frac{3(n_b-\ell_b)(n_b-\ell_b-1)\mu_b}{(n_b-\ell_b+\mu_b)(n_b-\ell_b+\mu_b-1)(n_b-\ell_b+\mu_b-2)}$$

$$M_0(\underline{\mu}, \underline{\ell}) = M_0(\mu_p, \ell_p) M_0(\mu_m, \ell_m)$$

Then (for $x_i = p_1, p_2, m_1, m_2$; $x = p, m$):

$$b_{x_i x_j}(\underline{\mu}, \underline{\ell}) = \frac{2\mu_{x_i}}{n_x-\ell_x} M_0(\underline{\mu}, \underline{\ell}) N_0(\mu_b, \ell_b)$$

* See "List of Symbols" for verbal definitions.

$$b_{x_i b}(\underline{\mu}, \underline{\ell}) = \frac{2\mu_{x_i}}{n_b - \ell_b} \left(\frac{1}{3}\right)^{\mu_p + \mu_m - 1} M_0(\underline{\mu}, \underline{\ell}) N_0(\mu_b, \ell_b)$$

$$b_{b x_i}(\underline{\mu}, \underline{\ell}) = \frac{2}{3} \frac{n_x - \ell_x - 2\mu_x}{n_x - \ell_x} \left(\frac{1}{3}\right)^{\mu_p + \mu_m} M_0(\underline{\mu}, \underline{\ell}) N_1(\mu_b, \ell_b)$$

$$b_{bb}(\underline{\mu}, \underline{\ell}) = \begin{cases} \frac{2}{n_b - \ell_b} N_1(\mu_b, \ell_b) & \mu_p = \mu_m = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$b_{p_i m_j}(\underline{\mu}, \underline{\ell}) = b_{m_i p_j}(\underline{\mu}, \underline{\ell}) = 0$$

$$d_{x_i}(\underline{\mu}, \underline{\ell}) = \frac{2\mu_{x_i}}{n_x - \ell_x} \left(\frac{1}{3}\right)^{\mu_p + \mu_m} M_0(\underline{\mu}, \underline{\ell}) N_1(\mu_b, \ell_b)$$

$$d_b(\underline{\mu}, \underline{\ell}) = \begin{cases} 0 & \mu_p + \mu_m = 0 \\ \left\{ \frac{3}{n_b - \ell_b} \left[1 - \left(\frac{1}{3}\right)^{\mu_p + \mu_m} - 2(\mu_p + \mu_m) \left(\frac{1}{3}\right)^{\mu_p + \mu_m} \right] N_0(\mu_b, \ell_b) \right. \\ \qquad \left. + \frac{2}{n_b - \ell_b} \left(\frac{1}{3}\right)^{\mu_p + \mu_m} N_1(\mu_b, \ell_b) \right\} M_0(\underline{\mu}, \underline{\ell}) \\ \qquad \mu_p + \mu_m > 0 \end{cases}$$

$$\lambda_{x_i}(t) = \lambda_{x_i} ; \lambda_b(t) = \lambda_b$$

Table 3.6b

SIFT INPUT PARAMETERS *

Case 1a: $\underline{L} = \underline{\ell} = (\ell_p, \ell_b)$ $\underline{\mu} = (\mu_p, \mu_b)$

$$b_{pp}(\underline{\mu}, \underline{\ell}) = \begin{cases} 1 & \underline{\mu} = (1, 0) \\ 0 & \text{otherwise} \end{cases}$$

$$b_{pb}(\underline{\mu}, \underline{\ell}) = b_{bp}(\underline{\mu}, \underline{\ell}) = 0$$

$$b_{bb}(\underline{\mu}, \underline{\ell}) = \begin{cases} 1 & \underline{\mu} = (0, 1) \\ 0 & \text{otherwise} \end{cases}$$

$$d_{xy}(\underline{\mu}, \underline{\ell}) = 0 \quad \text{all } x, y$$

$$\lambda_p(t) = \lambda_p \qquad \lambda_b(t) = \lambda_b$$

Cases 1b, 2 (two independent subsystems):

$\underline{L} = \underline{\ell} = \ell_p$	$\underline{L} = \underline{\ell} = \ell_b$
$b_{pp}(\mu, \ell) = \begin{cases} 1/s & \mu = 1 \\ 0 & \mu \neq 1 \end{cases}$	$b_{bb}(\mu, \ell) = \begin{cases} 1/s & \mu = 1 \\ 0 & \mu \neq 1 \end{cases}$
$d_{pp}(\mu, \ell) = 0$	$d_{bb}(\mu, \ell) = 0$
$\lambda_p(t) = \lambda_{0p} s$	$\lambda_b(t) = \lambda_{0b} s$
$s = 1 + r(1-p_{tr})$	(r = 0 for case 1b)

* See "List of Symbols" for verbal definitions.

Table 3.6b (Cont.)

Case 1c

Same as case 1a except:

$$b_{pb}(\underline{\mu}, \underline{\ell}) = \begin{cases} 1 & \mu = (1, 0) \\ 0 & \text{otherwise} \end{cases}$$

$$b_{bp}(\underline{\mu}, \underline{\ell}) = \begin{cases} 1 & \mu = (0, 1) \\ 0 & \text{otherwise} \end{cases}$$

creates two critical situations: a bp_j-critical fault and a p_ip_j-critical fault. Although such an event does not necessarily cause the system to fail, it was elected to treat all such events as fatal and hence to reflect their probabilities in the $d_x(\underline{\mu}, \underline{\ell})$ parameters. Since these are clearly events of relatively low probability, the added complexity needed to account for the possibility that the system could recover from them was not felt to be justified. Treating all such events as system failures, of course, again overbounds the true failure probability. The parameters $d_{p_i}(\underline{\mu}, \underline{\ell})$ and $d_{m_i}(\underline{\mu}, \underline{\ell})$ thus account for the event just described. The parameter $d_b(\underline{\mu}, \underline{\ell})$ is the probability either that at least two buses are used by memories or processors containing latent faults or that one bus and at least one memory or processor contains a latent fault and that, when a new bus fault occurs, it affects an active bus. Again both events produce a pair of critical fault situations.

The SIFT parameters shown in Table 3.6 are largely self-explanatory. The first three cases (cases 1a, 1b and 1c) differ only in the nature of the coupling between the two stages (cf. paragraph 3.2). The fourth case allows transients to occur at a rate r times the permanent fault rate. Since the probability of a "leaky" transient is $1 - p_{tr}$ and since leaky transients do not produce coverage failures, the probability that an arbitrary fault produces a critical fault situation is reduced by the probability $1/[1 + r(1-p_{tr})]$ that the fault is a leaky transient.

In addition to the parameters specified in Table 3.6, the CARE3 model must have access to the functions, defined in Table 3.5, used to characterize coverage. Since COVRGE has

not yet been implemented, these functions were generated by hand. These functions are easily defined in the permanent fault case:

$$p_1(t|\tau, x_i) = \begin{cases} e^{-\delta_{x_i}(t-\tau)} & \text{FTMP} \\ \left.\begin{array}{ll} 1 & 0 \le t-\tau \le \tau_0 \\ 0 & \text{otherwise} \end{array}\right\} & \text{SIFT} \end{cases}$$

$$p_2(t|\tau, x_i) = 0$$

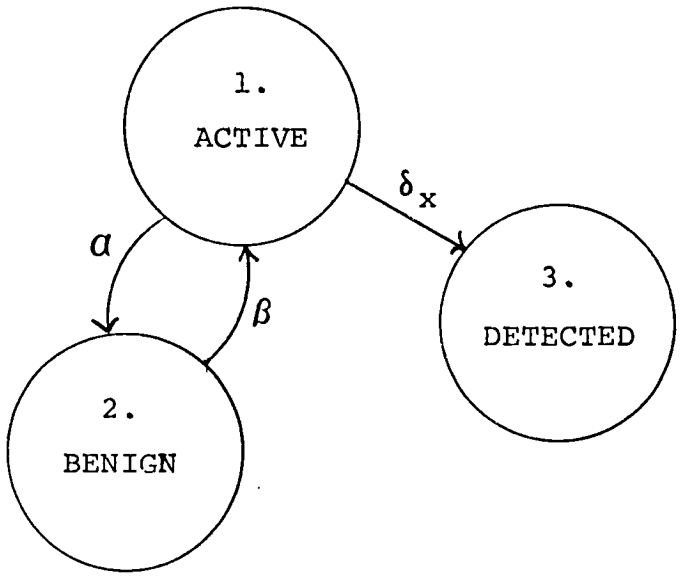
$$p(t|\tau, x_i, y_j) = 0$$

$$q(t|\tau, x_i, y_j) = \delta(t)$$

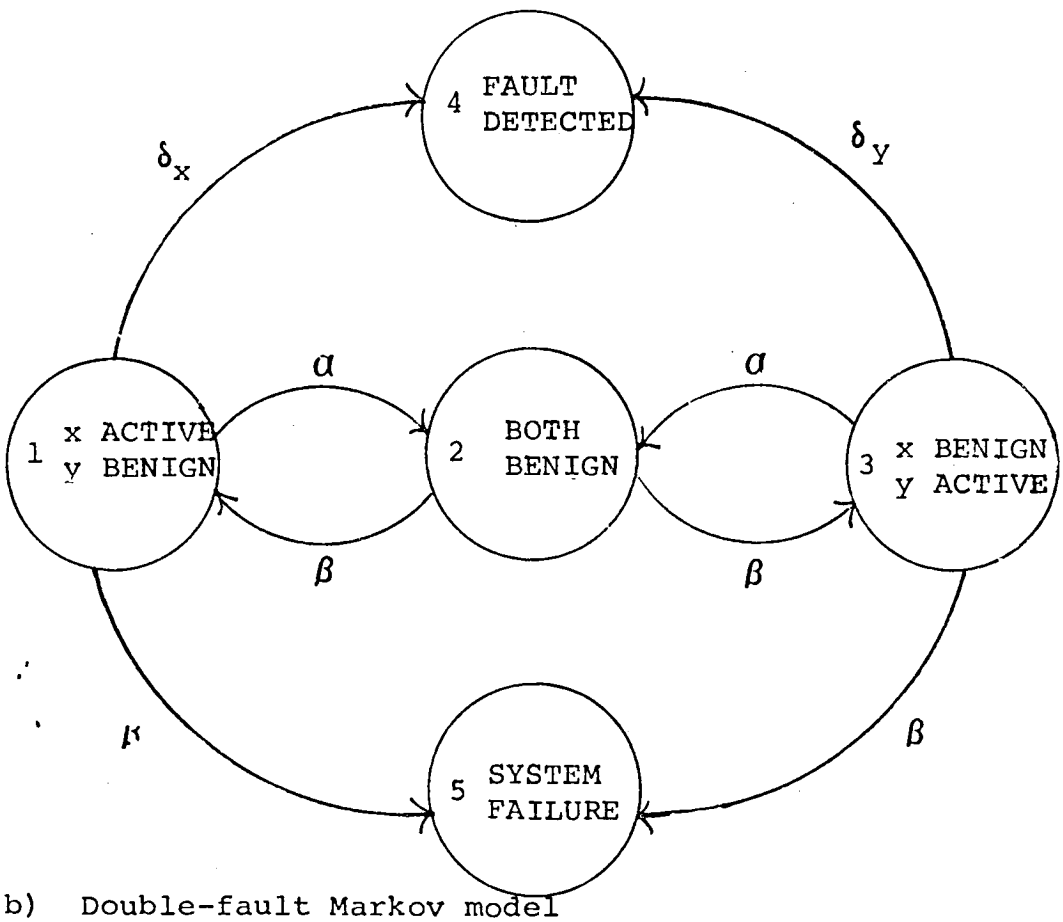
with δ_{x_i} the FTMP fault detection rate, τ_0 the SIFT detection delay, and $\delta(t)$ the Dirac delta function.

In the FTMP intermittent case, the first two of these functions are defined by a three-state Markov model and the last two by a five state Markov model, as shown in Figure 3.5. If $p_{ij}(t|\tau)$ represents the probability of being in state i at time t given that the system described by the three-state Markov model was in state j at time τ, and if $P_{ij}(t|\tau)$ is similarly defined for the five-state model, then

$$\begin{aligned} p_1(t|\tau, x) &= p_{11}(t|\tau) \\ p_2(t|\tau, x) &= p_{21}(t|\tau) \\ p(t|\tau, x, y) &= p_{11}(t|\tau) + p_{21}(t|\tau) + p_{31}(t|\tau) \\ q(t|\tau, x, y) &= \beta[p_{11}(t|\tau) + p_{31}(t|\tau)] \end{aligned}$$



a) Single-fault Markov model



b) Double-fault Markov model

Figure 3.5 INTERMITTENT FAULT MODEL

The functions $p_{ij}(t|\tau)$ and $P_{ij}(t|\tau)$ are readily determined either by hand (the first function involves solving a quadratic equation, the second a cubic) or by using one of the techniques described in Appendix 1.

3.4 PROGRAMMING APPROACHES FOR SYSTEM UNRELIABILITY MODEL

3.4.1 INTRODUCTION

The following paragraphs describe the techniques used to program the reliability model RM4 postulated in paragraph 3.1.4. For illustrative purposes, the parameters and dimensions discussed are those used for the FTMP model. As will become apparent, however, these parameters and dimensions can be readily modified as required to accommodate other situations.

3.4.2 COMPUTATION OF $Q_\ell(t)$ RECURSIVELY

In order to compute the probabilities $Q_\ell(t)$ recursively where $\ell \rightarrow (i, j, k)$†, an array must be defined for the $Q_\ell(t)$ probabilities so that $Q_{i-1, j, k}(t)$, $Q_{i, j-1, k}(t)$ and $Q_{i, j, k-1}(t)$ can be accessed when computing $Q_{i, j, k}(t)$.

If NP = no. of processors = 15; NPS = no. of processor survivors = 2

NM = no. of memories = 9; NMS = no. of memory survivors = 2

NB = no. of buses = 5; NBS = no. of bus survivors = 2

ITMAX = maximum no. of time steps = 50

QLT = array representing $Q_\ell(t)$, then

the array QLT must be dimensioned

---

†For purposes of this example, ℓ is a three-dimensional vector, $\ell = (i, j, k)$, with i denoting the number of failed processors, j the number of failed memory units and k the number of failed buses.

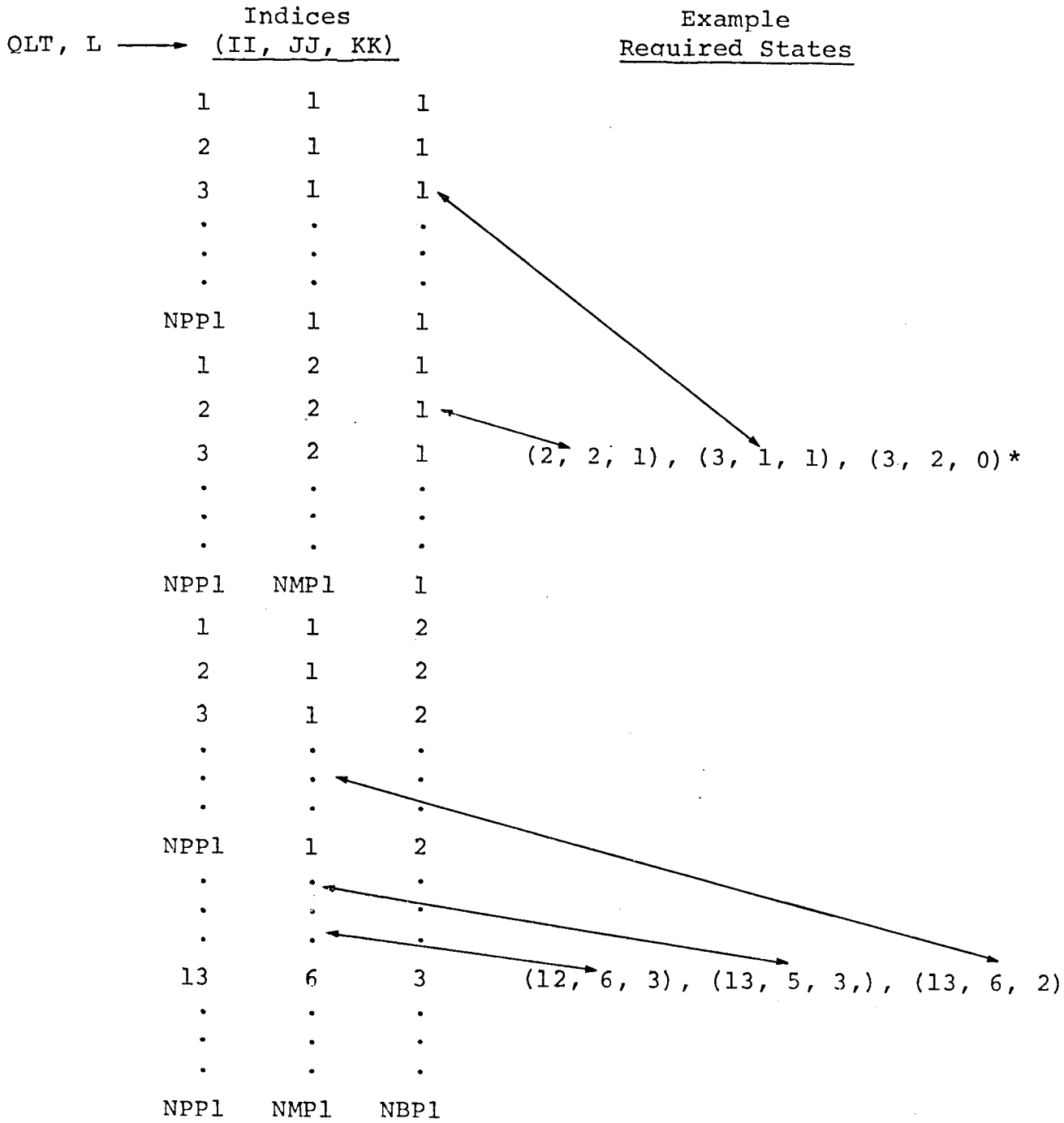
(NP - NPS + 1, NM - NMS + 1, NB - NBS + 1, ITMAX)=(14, 8, 4, 51), which includes 0 processor failures, 0 memory failures, 0 bus failures and time 0.

The immediate requirement then becomes the definition of a loop structure within the program for computing $Q_{\underline{\ell}}(t)$ so that all required probabilities have been previously computed and stored in the array. For example, when computing $Q_{\underline{\ell}}(t)$ for $\underline{\ell} \rightarrow (3, 2, 1)$, $Q_{2, 2, 1}(t)$, $Q_{3, 1, 1}(t)$ and $Q_{3, 2, 0}(t)$ must have been previously computed and stored in the QLT array.

Let II, JJ, KK, IT be the indices into the QLT array representing $Q_{i, j, k}(t)$. The basic structure in FORTRAN is then as shown on the following page.

```
C   Basic Fortran Algorithm
C
      NPP1 = NP + 1
      NMP1 = NM + 1
      NBP1 = NB + 1
      DO 100 KK = 1, NBP1
      DO 100 JJ = 1, NMP1
      DO 100 II = 1, NPP1
C
      I = IIM1 = II - 1
      J = JJM1 = JJ - 1
      K = KKM1 = KK - 1
      DO 75 IT = 1, ITMAX
C   Compute Q (II, JJ, KK, IT) using
C           Q (IIM1, JJ, KK, IT), Q (II, JJM1, KK, IT),
C           Q (II, JJ, KKM1, IT) where computing subroutines
C           use I, J and K
     75 Continue
C
      100 Continue
C
```

This structure would compute the state probabilities in the sequence as shown on the following page.



* A state vector with an index of 0 is defined as having 0 probability because a 0 index represents a negative component in the state vector (i, j, k), and hence designates a non-existent state.

Clearly all state probabilities will have been previously defined and stored in QLT array so that they are available when required.

Several problems occur if QLT is dimensioned and computed in this manner:

1. CDC Fortran Extended allows a maximum of 3 array declarators. Therefore the statement:

   DIMENSION QLT (14, 8, 4, 51)

   is an illegal declaration and will not compile.

2. The amount of memory required for such an array would be enormous:

   14 x 8 x 4 x 51 words, i.e., 22,848 words

3. Extending the model to include, for example, I/O modules would cause a problem because this would require an added dimension to the array (if such a dimension were legal). This would also increase the size of the QLT array even further.

4. Unnecessary computation of state probabilities would result--namely those which are so small that they have no affect upon the resultant probability. For example, the probability associated with state (13, 6, 3), i.e., 13 failed processors, 6 failed memory modules and 3 failed buses by time t may be too small to effect the system probability as a whole.

The solution to problem 1 is to create a mapping of $(i, j, k) \longrightarrow n$ which will reduce the QLT array to 2 dimensions: QLT(NMAX, IT). This will also solve problem 3; extending the

model from (i, j, k)⟶n to (i, j, k, m)⟶n would be a relatively minor programming enhancement. The only part of the program to change would be the mapping routine--plus model changes due to the addition of vector component m. This dimension solution, however, has no effect upon the size of the QLT array. The dimension statement now becomes DIMENSION QLT (448, 51) and would require the same amount of storage as previously.

The solution to problems 2 and 4 would be to modify the basic loop structure defined above so that:

a. The state probabilities are computed in a flow from largest to smallest; this enables the program to halt execution at a point where the probabilities no longer affect the result;

b. Only those probabilities actually needed to calculate the current state probability have to be stored in array QLT at any one time, thus reducing its size.

The following chart lists the computational flow required versus the basic computational flow. Each set consists of all permutations of vectors where the largest component of any vector is the set number. Vectors with components all less than the current set number were defined in previous sets; the probabilities associated with these vectors are not recomputed.

COMPUTATIONAL FLOW OF STATE VECTORS

CHART 1

BASIC COMPUTATIONAL FLOW			MODIFIED COMPUTATIONAL FLOW WITH SETS				EXAMPLE REQUIRED STATES
II	JJ	KK		II	JJ	KK	II-1 JJ KK, II JJ-1 KK, II JJ KK-1
1	1	1	Set 1	1	1	1	
2	1	1	Set 2	1	1	1*	
3	1	1		2	1	1	(1, 1, 1), (2, 0, 1)**, (2, 1, 0)**
4	1	1		1	2	1	
5	1	1		2	2	1	
6	1	1		1	1	2	
7	1	1		2	1	2	
8	1	1		1	2	2	
9	1	1		2	2	2	(1, 2, 2), (2, 1, 2), (2, 2, 1)
10	1	1	Set 3	1	1	1*	
11	1	1		2	1	1*	
12	1	1		3	1	1	
13	1	1		1	2	1*	
14	1	1		2	2	1*	
1	2	1		3	2	1	
2	2	1		1	3	1	
3	2	1		2	3	1	
.	.	.		3	3	1	
.	.	.		1	1	2*	
.	.	.		2	1	2*	
14	2	1		3	1	2	(2, 1, 2), (3, 0, 2)**, (3, 1, 1)
.	.	.		1	2	2*	
.	.	.		2	2	2*	
.	.	.		3	2	2*	
14	8	1		1	3	2	
1	1	2		2	3	2	
2	1	2		3	3	2	
.	.	.		.	.	.	
.	.	.		.	.	.	
.	.	.		.	.	.	
14	8	2		3	3	3	
.	.	.	Set 4	1	1	1*	
.	.	.		.	.	.	(3, 4, 4), (4, 3, 4), (4, 4, 3)
.	.	.		.	.	.	
14	8	4		.	.	.	
				4	4	4	
			Set 5	1	1	1*	
				.	.	.	
				.	.	.	
				.	.	.	
				5	5	4	
			Set 6	1	1	1*	
				.	.	.	
				.	.	.	
				.	.	.	
				6	6	4	

COMPUTATIONAL FLOW OF STATE VECTORS

CHART 1

MODIFIED COMPUTATIONAL FLOW WITH SETS

	II	JJ	KK
Set 7	1	1	1*
	2	1	1*
	3	1	1*
	4	1	1*
	5	1	1*
	6	1	1*
	7	1	1
	.	.	.
	.	.	.
	.	.	.
	7	7	4
	.	.	.
	.	.	.
	.	.	.
Set 14	1	1	1*
	.	.	.
	.	.	.
	.	.	.
	14	8	4

*These state probabilities have been previously computed and will not be recomputed. They are only dummy place holders used to show the algorithm more clearly.

**States with 0 indices do not exist.

The chart shows that only two sets need be in memory at any one time--the set being computed and its predecessor set. This occurs because the required states have either been computed in the predecessor set or previously in the set being computed. Also, with this method, only the state probabilities not computed in prior sets are stored in array QLT. Therefore, the number of unique states in each set for the case where

NP = 15, NPS = 2
NM = 9, NMS = 2
NB = 5, NBS = 2

is shown in the following chart:

Set	No. of Unique States	
1	1	
2	7	
3	19	
4	37	
5	36	
6	44	
7	52	largest two consecutive sets
8	60	
9	32	
10	32	
11	32	
12	32	
13	32	
14	32	

CHART 2

Set 7 and 8 are the largest two consecutive sets--having 52 and 60 states, respectively. Therefore, QLT array need only be dimensioned (112, 51), which is a total of 5712 words. Using this method, the amount of storage required for QLT array was decreased by 17,136 words.

The Fortran code required to compute the QLT array in sets, with only two sets of probabilities in memory at any one time follows:

```
C   FORTRAN ALGORITHM TO COMPUTE SETS OF STATES
C
C   Compute QLT (1, IT) for ℓ⟶(0, 0, 0)directly for all time
    steps.
    .
    .
    .
C   Initialize NSET(ISET) for set 1 to 1--only one state
    vector exists in set 1:  (0, 0, 0).
    NSET(1) = 1
C
C   Compute maximum number of failures permitted including 0
    NPF = NP - NPS + 1
    NMF = NM - NMS + 1
    NBF = NB - NBS + 1
C
C   Compute maximum indicies.
    NPP1 = NP + 1
    NMP1 = NM + 1
    NBP1 = NB + 1
C
C   Determine maximum set to compute.
    MAX = MAX0 (NPF, NMF, NBF)
```

```
C
C   Compute sets of state vector probabilities.
    DO 200 ISET = 2, MAX
    ISETB = ISET
    ISETM = ISET
    ISETP = ISET
    IF (ISETB.GT.NBP1) ISETB = NBP1
    IF (ISETM.GT.NMP1) ISETM = NMP1
    IF (ISETP.GT.NPP1) ISETP = NPP1
C
C   Initialize QLT index N to the number of vectors in the
C   previous set plus one.
    NUMPREV = NSET (ISET-1)
          N = NUMPREV + 1
    IF (ISET.EQ.2) GO TO 60
C
C   Pop vector probabilities off QLT array which were defined
C   two sets ago by moving the predecessor set up in the array.
    NPOP = NSET (ISET-2)
    DO 50 M = 1, NUMPREV
    MM = NPOP + M
C   Transfer QLT(MM, IT) for all time steps.
    DO 50 IT = 1, ITMAX
    QLT(M, IT) = QLT(MM, IT)
    50 CONTINUE
C
    60 Continue
C
C   Initialize unique state vector's counter to 0.
    NSTOT = 0
```

```
C
C   Begin main three loops which define the state vectors
    DO 100 KK = 1, ISETB
    DO 100 JJ = 1, ISTEM
    DO 100 II = 1, ISETP
C   Do not compute any previously computed state vector
C   probabilities.
    IF (II.LT.ISET.AND.JJ.LT.ISET.AND.KK.LT.ISET) GO TO 100
    I = II-1
    J = JJ-1
    K = KK-1
C
C   Compute QLT(N, IT) for all time steps.
    DO 75 IT=1, ITMAX
    .
    .
    .
    75 CONTINUE
C
C   Increase QLT index N and unique vector counter NSTOT by
C   one.
    N = N + 1
    NSTOT = NSTOT + 1
C
    100 CONTINUE
C
C   Store total number of unique vectors for the current set
C   ISET.
    NSET(ISET) = NSTOT
    200 CONTINUE
C
```

This Fortran structure is the basic programming core for the various CARE III models programmed thus far.

3.4.3 PROGRAM DIFFERENCES PER MODEL

The subroutine which computes the unique mathematical calculations for each model is subroutine SUMMAT. This subroutine and its associated functions vary for each model. They represent the numerator in the integrand of the integrated form of the Kolmogorov equation:

$$Q_\ell(t) = e^{-\int_0^t \lambda_\ell(\tau)\,d\tau} \int_0^t \frac{\overbrace{\left[\sum_j Q_j(\tau) + P_j(\tau)\bar{c}_{j\ell}(\tau)\right]\lambda_{j\ell}(\tau)}^{\text{SUMMAT}}}{e^{-\int_0^\tau \lambda_\ell(\eta)\,d\eta}}\,d\tau$$

The main concern in programming subroutine SUMMAT for each model is to eliminate redundant computations. Two types of function computations are required: functions which are time dependent and functions which are vector dependent; i.e., dependent upon (i, j, k). The time dependent functions must be removed from subroutine SUMMAT and computed in subroutine TDEPEND. TDEPEND computes all time dependent functions once and stores them in arrays. These arrays can later be accessed from subroutine SUMMAT each time the vector changes. This approach keeps execution time at a minimum because it takes much less time to retrieve a function value from an array than it does to recompute the function each time the vector (i, j, k) changes. SUMMAT then computes the vector dependent portions of the model while accessing the time dependent arrays.

3.4.4 NUMERICAL INTEGRATION TECHNIQUES

The Trapezoidal rule

$$\int_{x_0}^{x_1} f(x)dx = \frac{\Delta x}{2} [f(x_0) + f(x_1)]$$

and Simpson's 1/3 rule

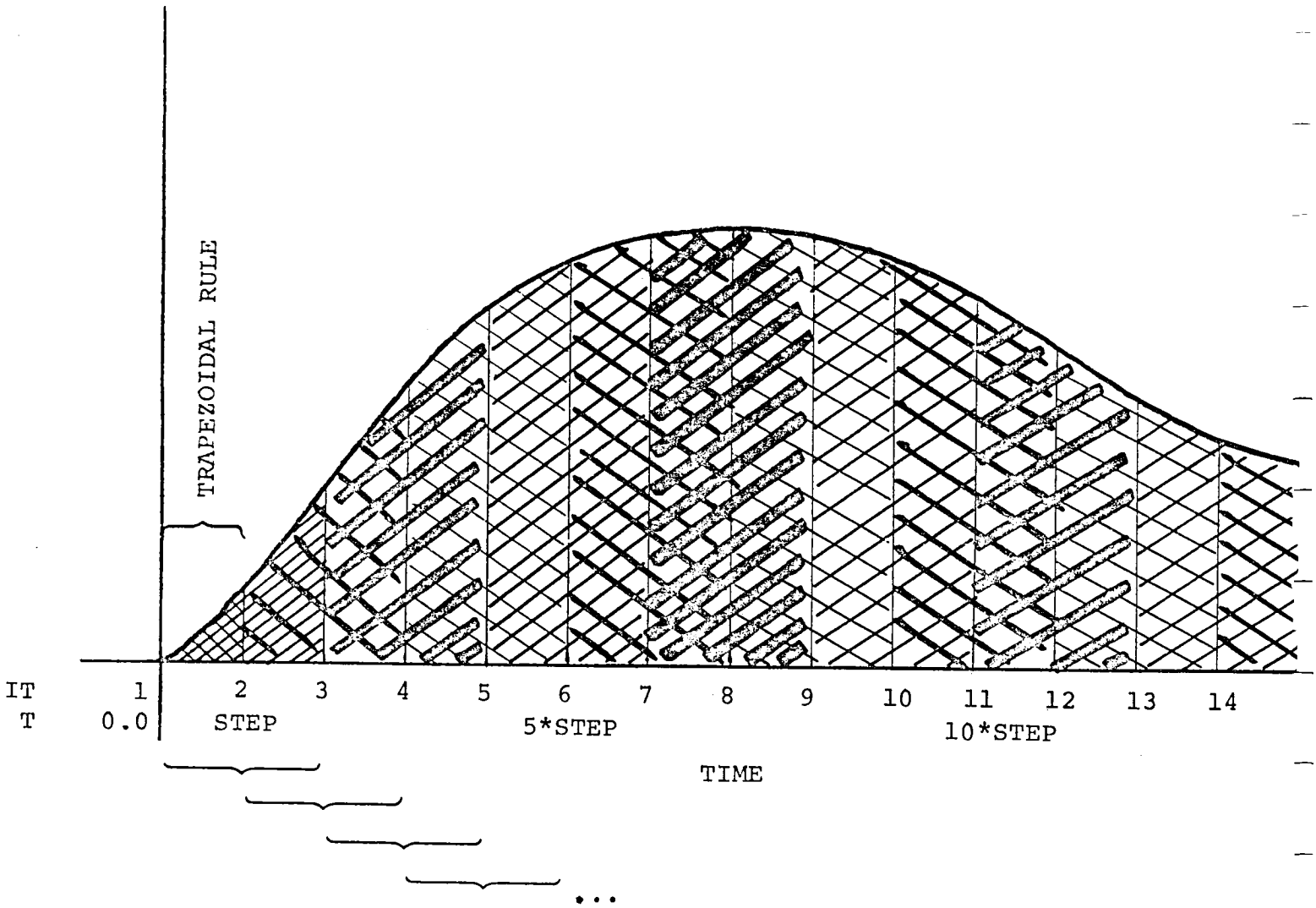
$$\int_{x_0}^{x_2} f(x)dx = \frac{\Delta x}{3} [f(x_0) + 4f(x_1) + f(x_2)]$$

are the numerical integration techniques used within the program to compute the integral

$$\int_0^t \frac{\left[\sum_j O_j(\tau) + P_j(\tau)\bar{c}_{j\ell}(\tau)\right] \lambda_{j\ell}(\tau)}{e^{-\int_0^\tau \lambda_\ell(\eta)d\eta}} d\tau$$

of the Kolmogorov equation.

The Trapezoidal rule is used to compute the integral from time 0 to time STEP where STEP is the step size or Δx. Simpson's 1/3 rule is used to compute the remaining intervals as shown in the following Figure 3.6. (The subroutines associated with these numerical techniques are called TRAPINT and SIMPINT.)



Simpson's 1/3 Rule

$$\underline{IT}$$

where $QLT(N, 1) = 0.0$

$$QLT(N, 2) = \text{area } 1 * e^{-\int_0^t \lambda_\ell(\tau)\,d\tau}$$

$$QLT(N, 3) = \text{AREA } 2 * e^{-\int_0^t \lambda_\ell(\tau)\,d\tau}$$

$$QLT(N, 4) = (\text{AREA } 3 + \text{area } 1) * e^{-\int_0^t \lambda_\ell(\tau)\,d\tau}$$

$$QLT(N, 5) = (\text{AREA } 4 + \text{AREA } 2) * e^{-\int_0^t \lambda_\ell(\tau)\,d\tau}$$

$$\vdots$$

$$QLT(N, ITSTPS) = (\text{area ITSTPS} - 1 + \text{area ITSTPS} - 3 + \text{area ITSTPS} - 5 + \ldots + \text{area } 1)\ e^{-\int_0^t \lambda_\ell(\tau)\,d\tau}$$

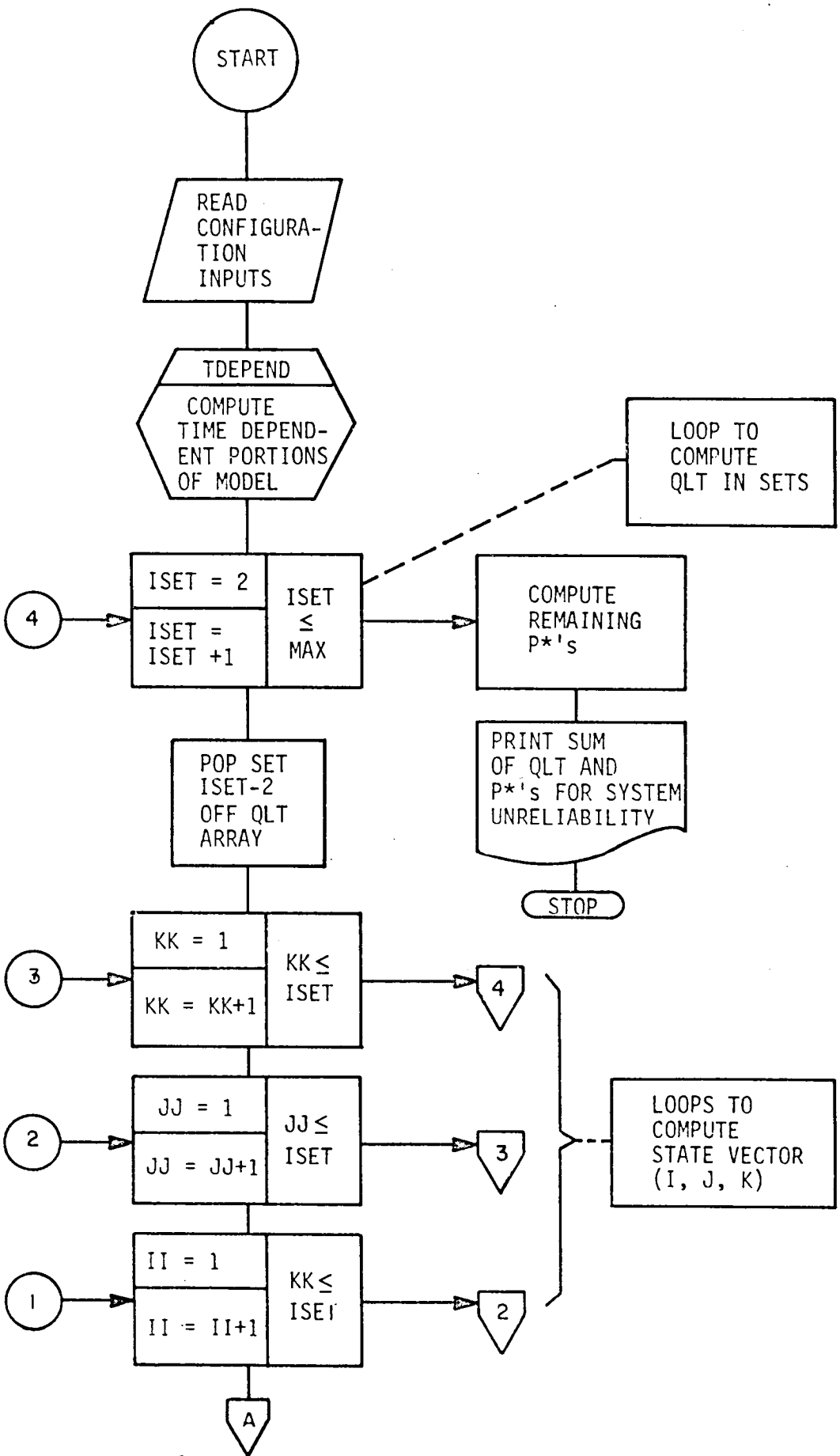
INTEGRATION METHODS

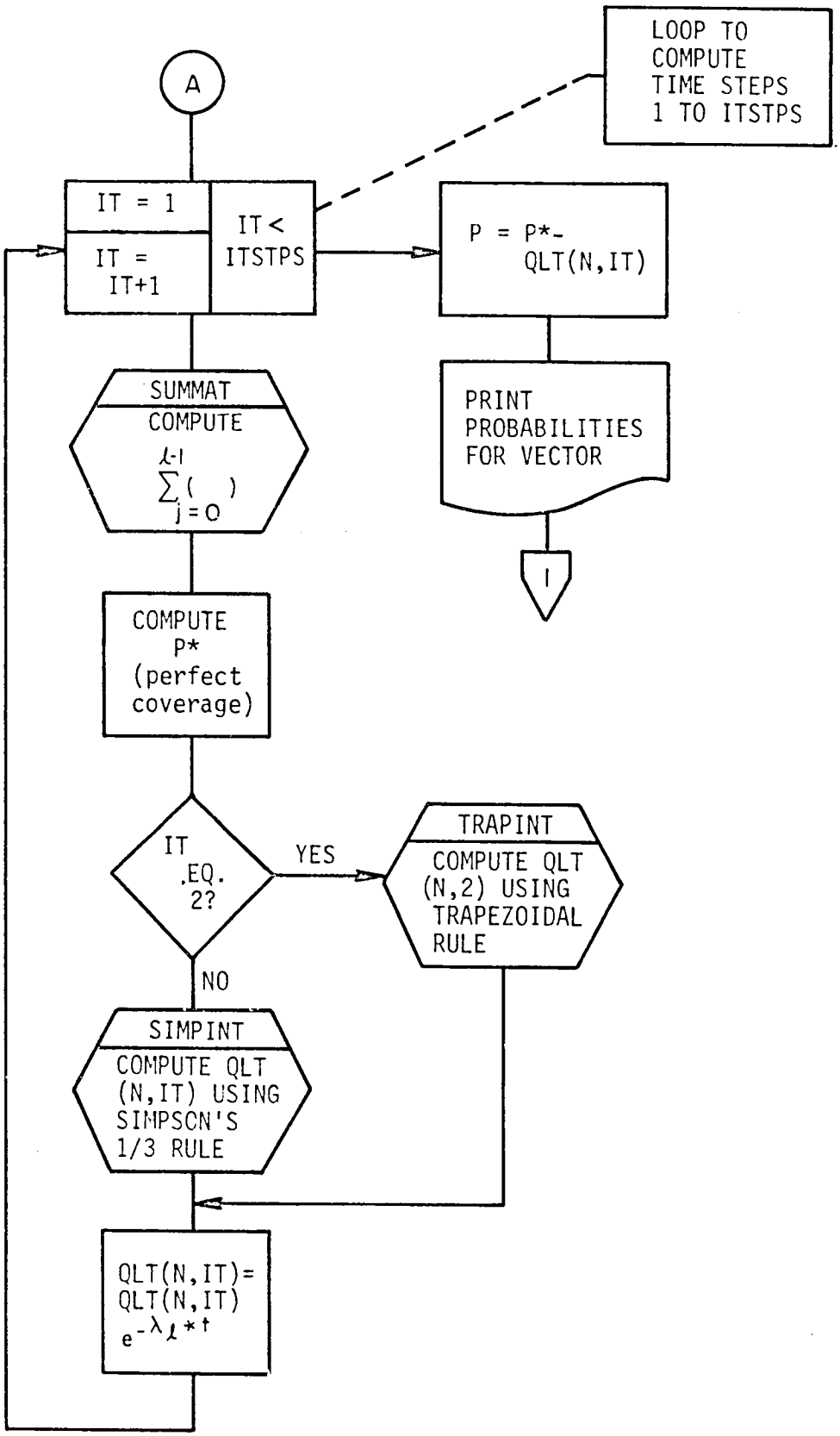
Figure 3.6

3.4.5 MACRO FLOW CHART OF SYSTEM UNRELIABILITY MODEL

The following macro flow chart shows the organization of the entire basic model which computes the system unreliability. The loop structure computing the vectors in sets is shown in relationship to the subroutines TDEPEND, SUMMAT, TRAPINT and SIMPINT.

MACRO FLOW CHART





PAGE 2 OF 2

CARE 3 MACRO FLOW CHART

4.0 CARE III PROGRAM STRUCTURE

An implementation of a Modularized Direct Access Information System is the proposed structure for the CARE III system. The system will consist of three main modules:

a. Batch or interactive input processor: CAREINB or CAREINI

b. Coverage model: COVRGE

c. Reliability model: CARE3

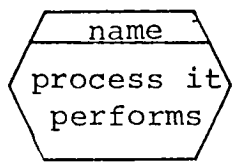
The following flow diagrams depict the proposed design of the CARE III system.

Two text input files are required: one to define the computer configuration and one to aid in the calculation of the coverage model. If coverage is preset per stage in the configuration file INFILE, the coverage input file CVFILE need not be defined by the user.

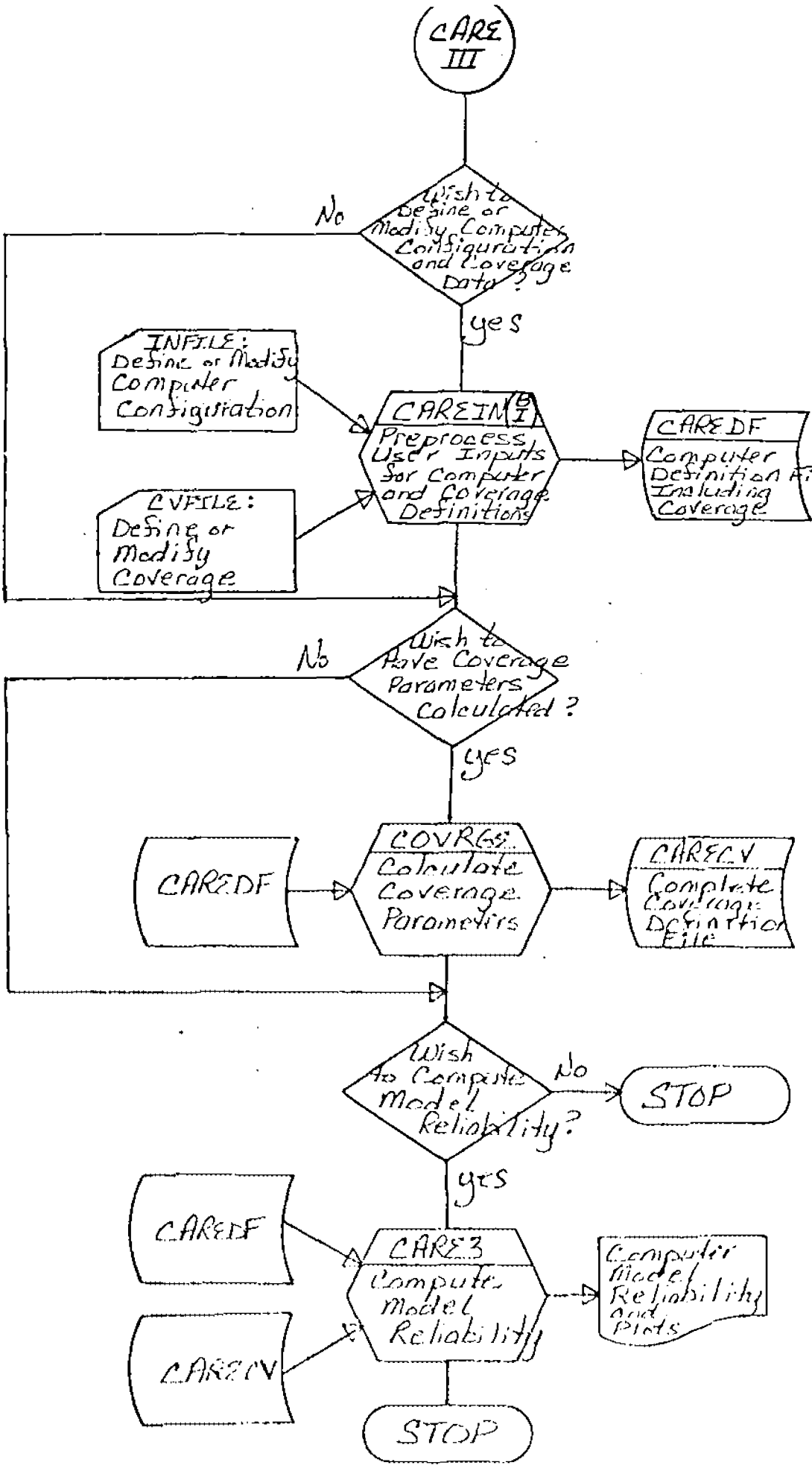
The Direct Access Information System (DAIS) files generated by CARE III are designed to be random, word addressable mass storage files. Each record within these files can be accessed with a master index or subindex(es). The DAIS files will contain the processed user input required by programs COVRGE and CARE3. They will be made permanent disk files by CARE III so that they can be modified if desired without having to reinput the entire data set. Thus a second run can use existing files CAREDF and CARECV, after minor modifications have been made to them, by running program CAREIN using only an updated portion of the original input. This capability is especially convenient if the user runs the interactive CAREIN program.

The DAIS files are to be created and accessed through the use of FORTRAN Mass Storage Input/Output (MSIO) subroutines OPENMS, WRITMS, READMS and CLOSMS. Record Manager word addressable file organization is used to implement these files.

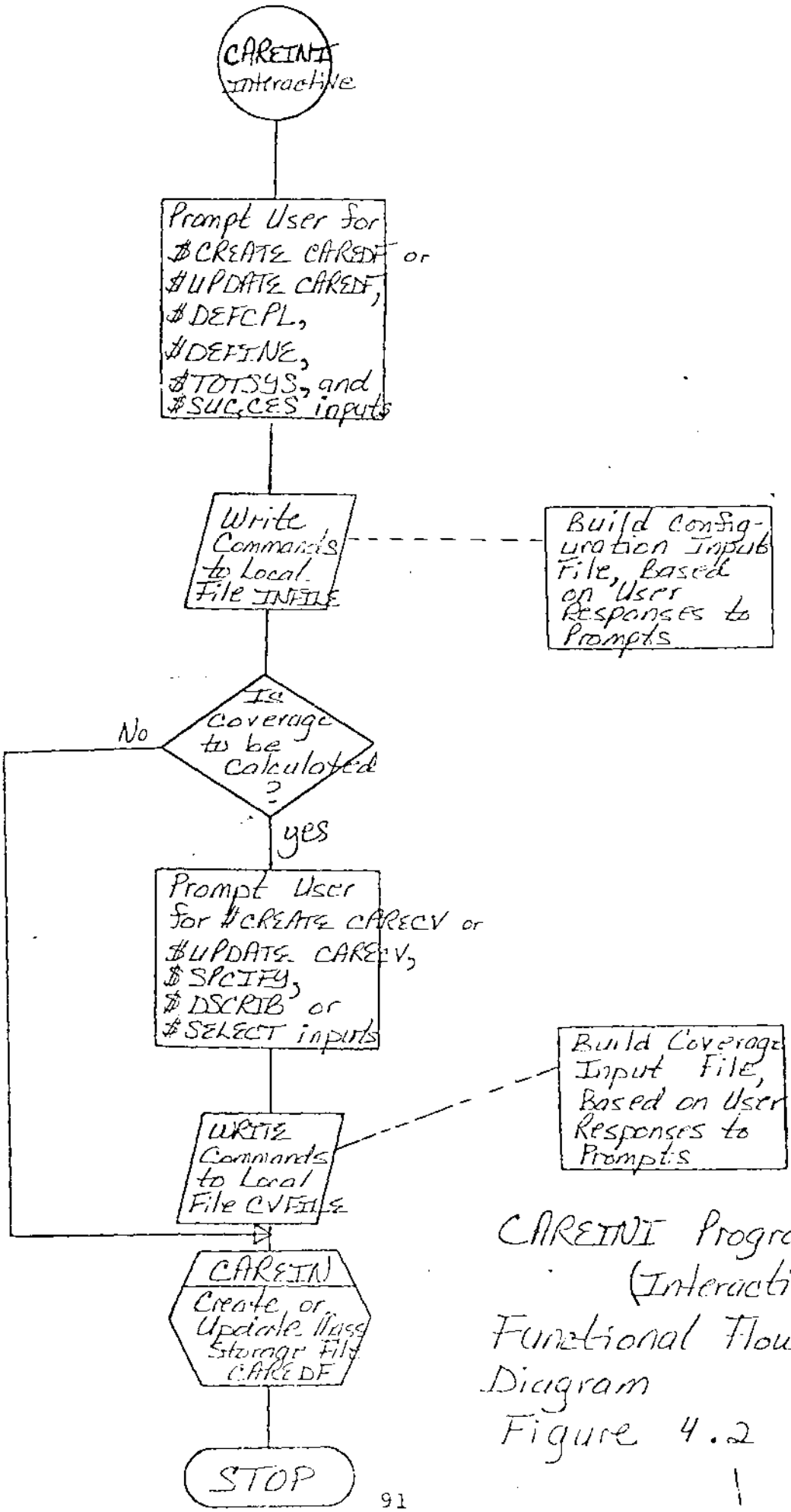
In the following flow diagrams the symbol



denotes a separate routine for which a separate flow diagram exists in the pages following. For a more detailed look at the proposed system, see the CARE III Computer Program Requirements Document.

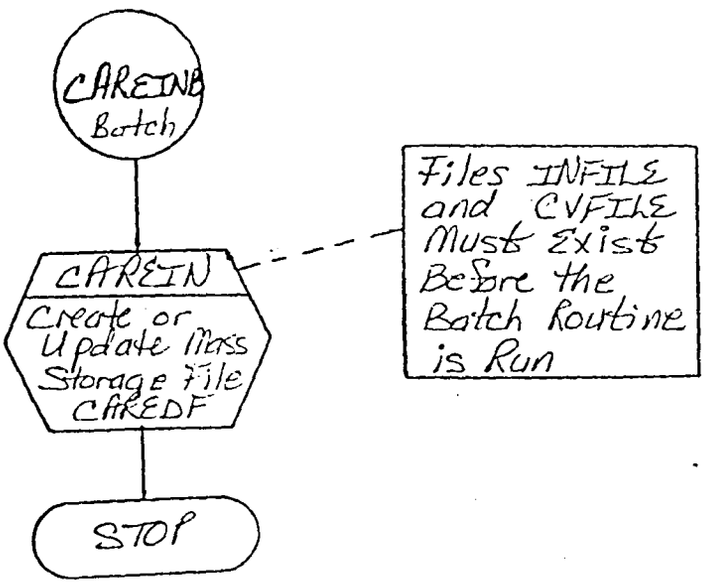


CARE III System Functional Flow-Diagram
Figure 4.1

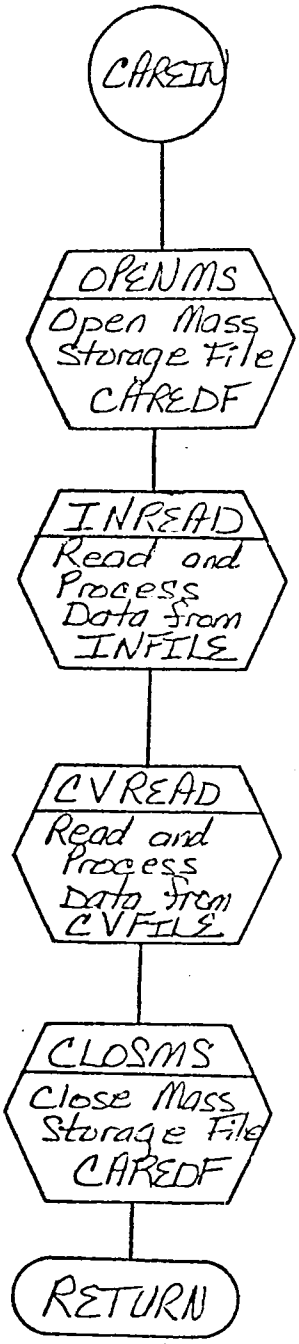


CAREINI Program (Interactive) Functional Flow-Diagram

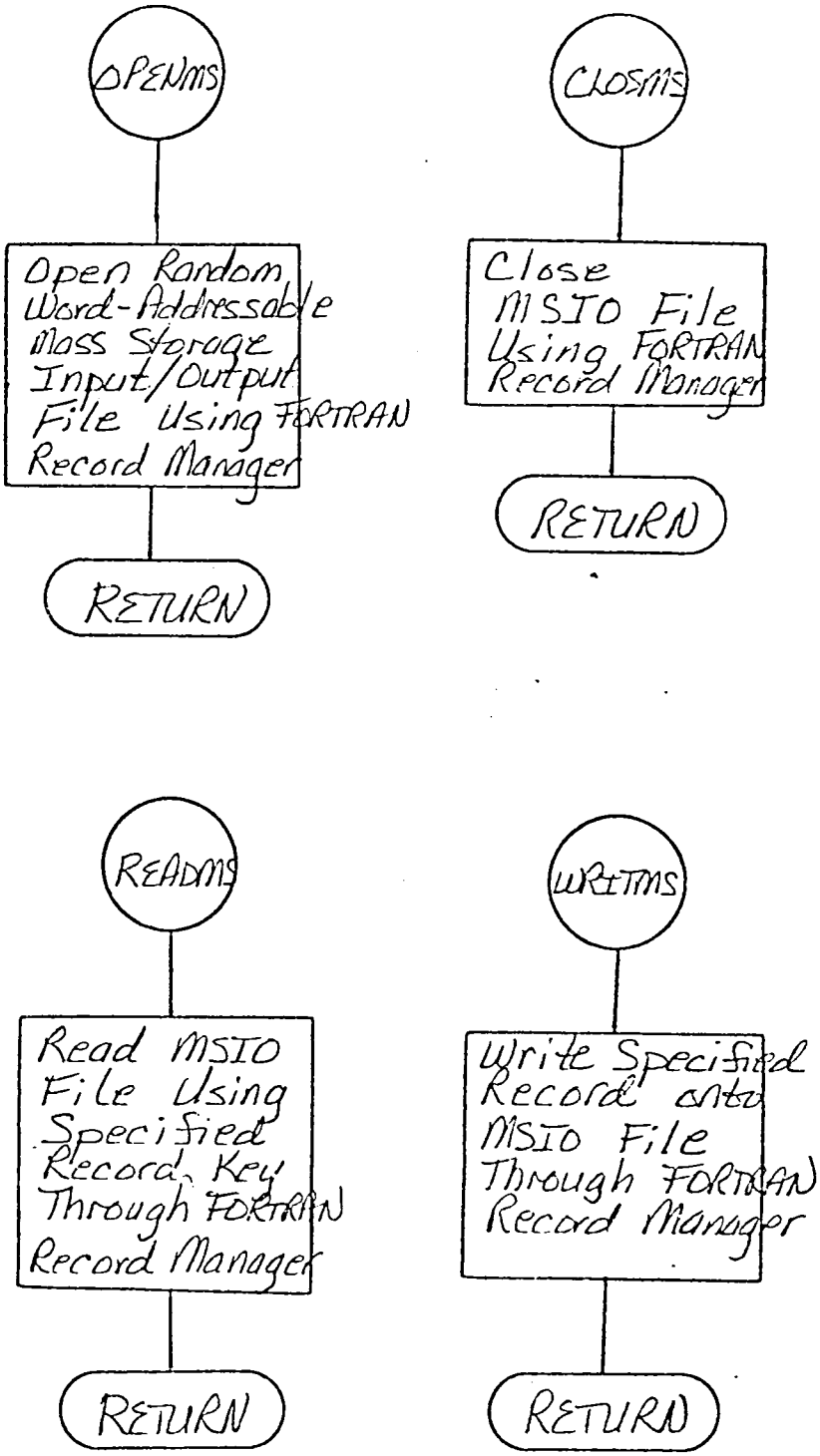
Figure 4.2



CAREINB Program
(Batch)
Functional Flow-
Diagram
Figure 4.3

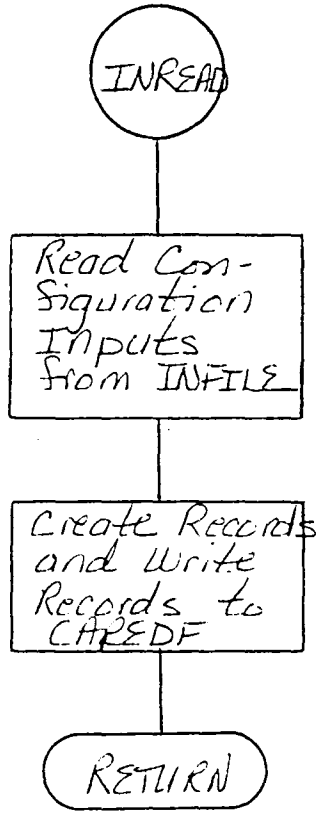


CAREIN Subroutine Functional Flow-Diagram
Figure 4.4

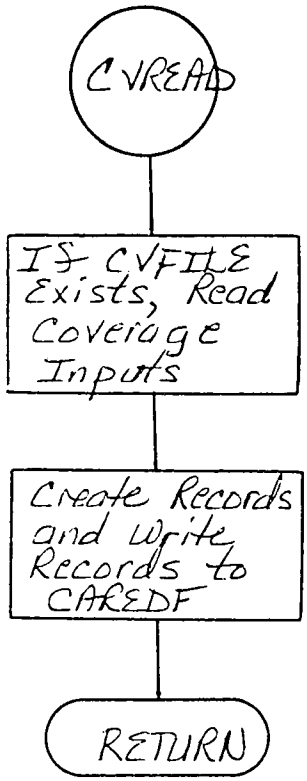


FORTRAN Library Routines: OPENMS
CLOSMS
READMS
WRITMS

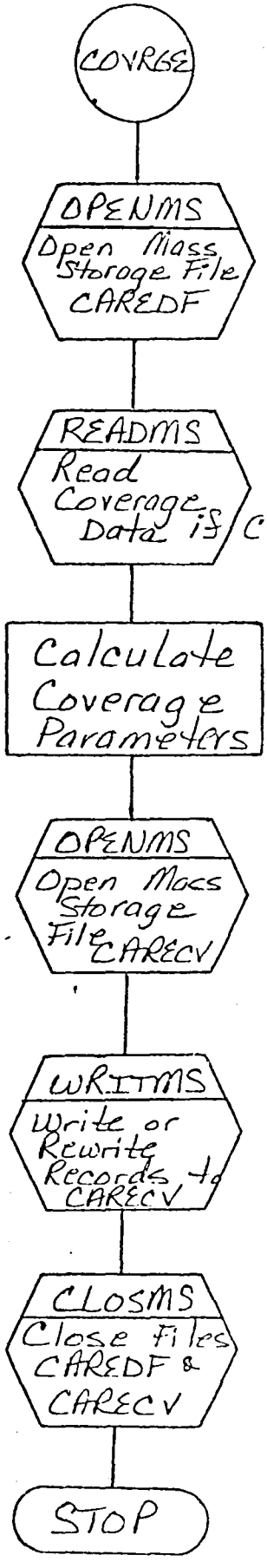
Figure 4.5



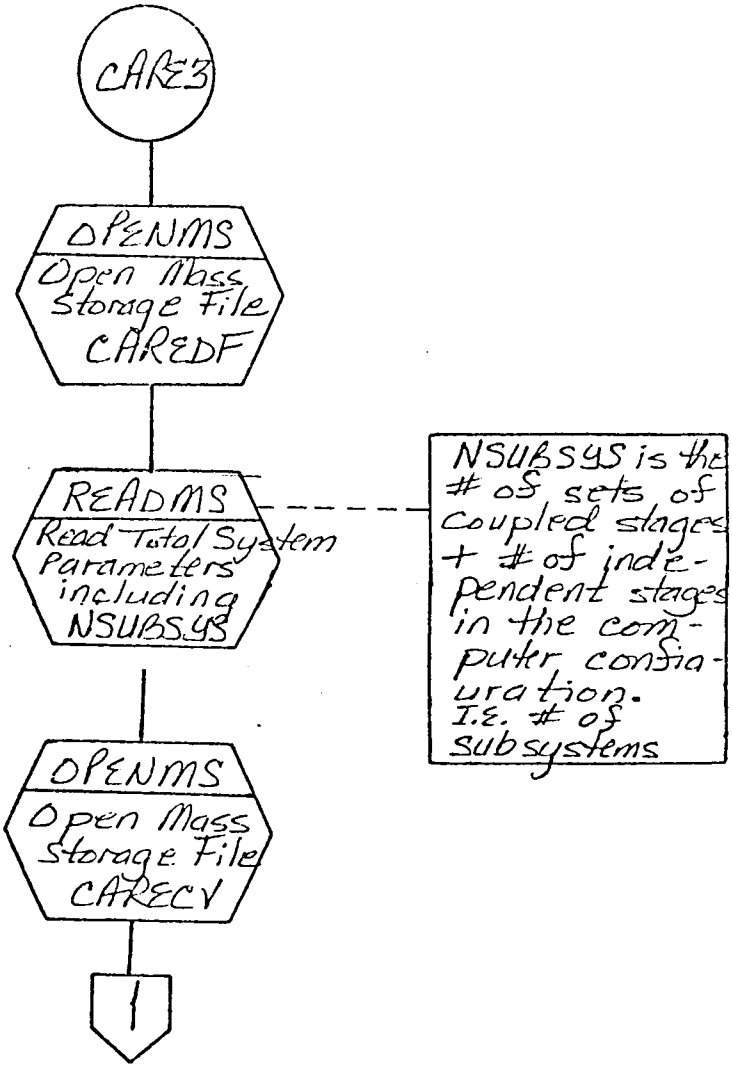
INREAD Subroutine
Macro Flow-Diagram
Figure 4.6



CVREAD Subroutine
Macro Flow-Diagram
Figure 4.7



COVRGE Program Functional Flow-Diagram
Figure 4.8



CARE3 Program Functional Flow-Diagram
Figure 4.9

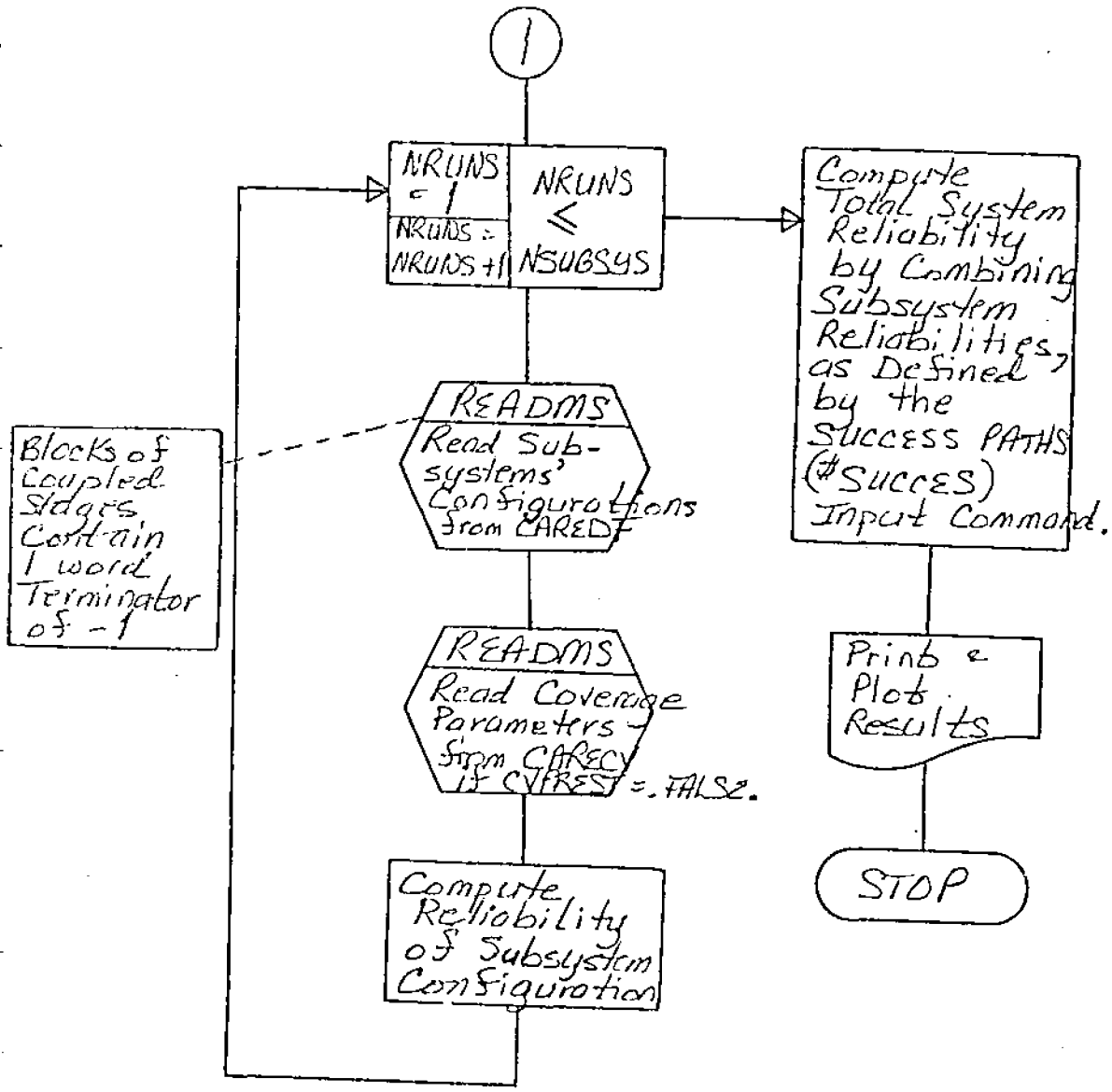


Figure 4.9 continued

REFERENCES

1. ----------, "An Engineering Treatise on the CARE II Dual Mode and Coverage Models," Final Report, NASA Contract L-18084A, November, 1975.

2. Bjurman, B. E., et. al., "Airborne Advanced Reconfigurable Computer Systems (ARCS)," Final Report, NASA Contract NAS1-13654, August, 1976.

3. Wensley, J. H., et. al., "Design Study of Software Implemented Fault Tolerance (SIFT) Computer," Interim Technical Report No. 1, NASA Contract NAS1-13792.

4. Smith, T. B., et. al., "A Fault-Tolerant Multiprocessor Architecture for Aircraft," Interim Report, NASA Contract NAS1-13782.

5. ----------, "Brassboard Fault Tolerant Spaceborne Computer (BFTSC," Final Report, USAF Contract 04701-75-C-0149, December, 1978.